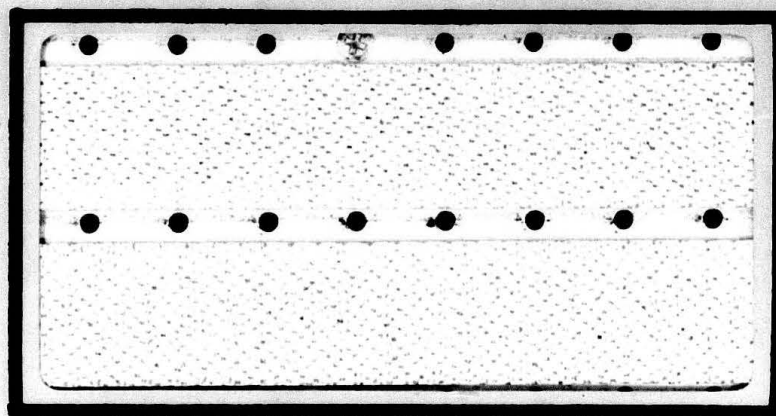


AD A125405



DTIC

FEB 23 1983

H

DTIC FILE COPY

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

83 02 022 094

AFIT/GE/EE/82D-64

DTIC
ELECTE
FEB 23 1983
S H

ANALYSIS OF SIX ALGORITHMS FOR
BEARINGS-ONLY RANGING IN AN
AIR-TO-AIR ENVIRONMENT

THESIS

AFIT/GE/EE/82D-64

DONALD R. STIFFLER
Captain USAF

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

Approved for public release; distribution unlimited

AFIT/GE/EE/82D-64

ANALYSIS OF SIX ALGORITHMS FOR BEARINGS-ONLY
RANGING IN AN AIR-TO-AIR ENVIRONMENT

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

DONALD R. STIFFLER, B.S.E.E

Captain USAF

Graduate Engineering

December 1982

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special

Approved for public release; distribution unlimited



ACKNOWLEDGEMENTS

I owe a debt of thanks to many persons for helping this "dumb fighter pilot" produce a thesis which fulfills the requirements for a Masters degree and hopefully provides some information which may be useful to the Air Force. First, I would like to thank my wife, [REDACTED] for her understanding and encouragement. Without her support, the last six months would certainly have been a miserable experience. My thanks also to my thesis advisor, Dr. Peter Maybeck, without whose instruction and cheerfully given advice and suggestions a thesis on this particular topic would have been impossible. Finally, I wish to thank my typist, Mary Minnick, for her professional quality work.

TABLE OF CONTENTS

<u>Title</u>	<u>Page</u>
Acknowledgements	ii
List of Figures	v
List of Tables	vi
Abstract	vii
I Introduction	1
1.1 Background	1
1.2 Problem Statement and Objectives	2
1.3 Organization	3
II The Kalman Filter	4
2.1 Introduction	4
2.2 System Modelling	4
2.3 The Basic Kalman Filter	6
2.4 The Extended Kalman Filter	9
III Algorithm Development	12
3.1 Introduction	12
3.2 Reference Frames and Notation	13
3.2.1 Reference Frames	13
3.2.2 Notation	17
3.3 Algorithm Description	20
3.3.1 Overview	20
3.3.2 Algorithm One	23
3.3.3 Algorithm Two	30
3.3.4 Algorithm Three	36
3.3.5 Algorithm Four	38
3.3.6 Algorithm Five	42
3.3.7 Algorithm Six	44
3.4 Determination of Values for $\underline{Q}(t)$, $\underline{R}(t_1)$ and τ	46
3.4.1 Determination of $\underline{Q}(t)$ and τ for Algorithms One and Two	46
3.4.2 Determination of $\underline{Q}(t)$ for Algorithms Three Through Six	48
3.4.3 Determination of $\underline{R}(t_1)$	49
IV Method of Evaluation	51
4.1 The Monte Carlo Simulation	51
4.2 Sampling Rates and Sensor Noises Tested	54
4.3 Figures of Merit	56

<u>Title</u>	<u>Page</u>
V Results	58
5.1 Introduction	58
5.2 Maneuver Profiles	58
5.2.1 Fighter Profile	58
5.2.2 Target Profiles	61
5.3 Evaluation of Algorithm One and Two	62
5.4 Evaluation of Algorithm Three	64
5.5 Evaluation of Algorithm Four	67
5.6 Evaluation of Algorithm Five and Six	68
5.6.1 Introduction	68
5.6.2 Filter Convergence Enhancement	69
5.6.3 Algorithm Performance Against Non- Maneuvering Targets	74
5.6.4 Algorithm Performance Against Maneuvering Targets	79
5.6.5 Effects of Varying Measurement Errors ..	82
VI Conclusions and Recommendations	86
6.1 Conclusions	86
6.2 Recommendations	87
Bibliography	89
Appendix A: Additional Matrices for Algorithm One ...	91
Appendix B: Additional Matrices for Algorithm Two ...	93
Appendix C: Additional Matrices for Algorithm Three	98
Appendix D: Additional Matrices for Algorithm Four	99
Appendix E: Additional Matrices for Algorithm Five	101
Appendix F: Additional Matrices for Algorithm Six ...	103
Appendix G: Graphical Results for Algorithm One	104
Appendix H: Graphical Results for Algorithm Two	109
Appendix I: Graphical Results for Algorithm Three ...	119
Appendix J: Graphical Results for Algorithm Four	126
Appendix K: Graphical Results of Adding Pseudonoise to the Range Channel of Algorithm Five ..	135
Appendix L: Graphical Results of Algorithm Five	140
Appendix M: Graphical Results of Algorithm Six	171
Appendix N: Graphical Results of Algorithm Five Against Maneuvering Targets	201
Appendix O: Graphical Results of Algorithm Five Against Non-Maneuvering Targets With Varying Measurement Error Variances	226
Appendix P: Values for $\underline{P}(t_0)$	259
Appendix Q: Discussion of the Bayesian Filtering Approach	260
Vita	262

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3-1	Line of Sight Reference Frame	16
3-2	Spherical Reference Frame	18
3-3	Rotation Rate in the LOS Frame	25
3-4	Target Position in the Relative NED Frame	33
3-5	Algorithm Four Horizontal Intercept Geometry	40
4-1	Performance Evaluation	52
4-2	Example of SOFEPL Output	55
5-1	Fighter Maneuver Profile	59
G-1 through G-4	Algorithm One Performance	105-108
H-1 through H-9	Algorithm Two Performance	110-118
I-1 through I-6	Algorithm Three Performance	120-125
J-1 through J-8	Algorithm Four Performance	127-134
K-1 through K-4	Results of Adding Pseudonoise to the Range Channel of Algorithm Five .	136-139
L-1 through L-29	Algorithm Five Performance	141-170
M-1 through M-29	Algorithm Six Performance	172-200
N-1 through N-24	Algorithm Five Performance Against Maneuvering Targets	202-225
C-1 through O-32	Algorithm Five Performance with Varying Measurement Error Vari- ances	227-253

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
3-1	Notation Summary	19
5-1	Target Maneuver Profiles	60
5-2	Algorithm Five Performance, Nominal Error Variances	75
5-3	Algorithm Six Performance, Nominal Error Variances	75
5-4	Algorithm Five Performance, Target Profile 5, Varying Measurement Error Strengths	84
5-5	Algorithm Five Performance, Target Profile 7, Varying Measurement Error Strengths	84
P-1	Values for $P(t_0)$	259

✓ Six algorithms for finding the range to an airborne target using noise-corrupted bearing and bearing rate measurements from an airborne sensor were developed and evaluated. The algorithms all employ extended Kalman filters, one using a spherical coordinate system representation, and the other five using Cartesian system representation. The target models include independent, Gauss-Markov acceleration (two algorithms), constant turn rate motion in a single plane (one algorithm), and basically straight and level flight (three algorithms).

The algorithms were evaluated in a Monte Carlo analysis. The truth model for this analysis was based upon two trajectories, one for the target and one for the attacker, which were deterministically set before the Monte Carlo runs. The output of the simulation was composed of plots of the error process, including the mean of the errors, the mean of the errors plus and minus one standard deviation, and the positive and negative filter-generated estimate of the standard deviation of the errors.

It was found that, due to an observability problem, only the algorithms which modelled target motion as straight and level flight worked satisfactorily. These algorithms worked best in scenarios in which the actual target motion was nearly directly toward or directly away from the attacking aircraft.

ANALYSIS OF SIX ALGORITHMS FOR BEARING-ONLY RANGING IN AN AIR-TO-AIR ENVIRONMENT

CHAPTER I

INTRODUCTION

1.1 Background

The problem of determining the distance to a target when given only noise-corrupted bearing information has received increased attention in recent years (1; 2; 3; 4; 11). Throughout the 1970's, research has been conducted in this area by several individuals and organizations, most notably the U.S. Naval Underwater Systems Center in Newport, Rhode Island. Because the applications of bearing-only ranging have been limited chiefly to naval scenarios in which bearing measurements are obtained from acoustic sensors, however, the dynamics of the problem have been modelled as occurring slowly and in a single plane of motion.

Presently, the Air Force relies almost exclusively on radar for target acquisition and tracking. Because radar must actively transmit an RF signal in order to obtain target information, however, it is a relatively simple matter for the target to detect that he is being illuminated by a radar transmitter. The current trend toward "stealth" technology points toward the development of a sensor which could detect and track the line of sight to a target, providing accurate bearing and bearing rate measurements to the tracking algorithm. In order for this sensor to work effectively, however, it

would be necessary for the algorithm to compute the target's range from these angle and angle rate measurements. Thus one may see a future requirement for a bearing-only ranging algorithm which functions in the dynamic, three-dimensional environment of air-to-air combat.

1.2 Problem Statement and Objectives

This thesis presents the development and evaluation of six proposed algorithms for finding range to a target, given noise corrupted azimuth, elevation, azimuth rate, and elevation rate from a simulated sensor. The six algorithms proposed for this study all use the extended Kalman filter (6:44-59) to estimate range recursively. The evaluation of these algorithms is done by means of Monte Carlo simulations (5:325-329; 9) which provide a means of measuring algorithm performance under the conditions of specific combinations of sensor noises and engagement scenarios. Obviously, the six ranging methods cannot be evaluated under every possible combination of target and own-ship maneuver; therefore, only the most promising algorithms (based on tracking performance in one engagement scenario) are tested extensively for the following target trajectories:

- 1) benign target maneuvers which meet the target modelling assumption of the algorithm,
- 2) target maneuvers which are representative of real world situations,
- 3) target maneuvers which are expected to cause the greatest errors in the algorithm's range estimate.

In comparing the performance of the different algorithms, two main characteristics will be examined;

- 1) the time the algorithm takes to converge from erroneous initial conditions,
- 2) the tracking error after convergence.

1.3 Organization

This thesis is organized so that the reader can follow the development and testing in the chronological order in which they would logically occur. Chapter 1 contains the background information and states the problem and objectives of this study. Chapter 2 gives the reader unfamiliar with Kalman filtering techniques a short summary of the equations and assumptions involved. In Chapter 3 the development of the six algorithms to be tested is presented. Chapter 4 discusses the truth model and the method of evaluation. Chapter 5 contains the results of the study, and Chapter 6 contains the conclusions and recommendations.

CHAPTER II

THE KALMAN FILTER

2.1 Introduction

Because the ranging algorithms examined in this thesis are dependent upon some form of Kalman filter for range estimation, it is necessary that the reader be familiar, if not with the derivation, at least with the equations and basic assumptions associated with Kalman filters. This chapter presents a summary of the modelling assumptions and equations for both the basic Kalman filter, used with linear systems, and the extended Kalman filter (EKF), used with systems which have nonlinear dynamics or measurement equations.

Those readers familiar with Kalman filtering techniques may skip this chapter without loss of continuity. Those requiring a more in-depth description of stochastic processes and the derivation of the Kalman filter are referred to References 5 and 6.

2.2 System Modelling

The Kalman filter is designed to provide an estimate of the states of a system which can be modelled as n coupled first-order stochastic differential equations driven by white Gaussian noise, where the filter is provided with m measurements which are functions of the system states corrupted by zero-mean, white, Gaussian noise. A Kalman filter formulation to process continuous time measurements is possible; however, since all algorithms examined in this thesis

use sampled-data measurements appropriate for filter implementation in a digital computer, only the discrete-time formulation will be discussed here.

If the system dynamics and measurement equations are linear, they may be written

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) + \underline{G}(t)\underline{w}(t) \quad (1)$$

$$\underline{z}(t_i) = \underline{H}(t_i)\underline{x}(t_i) + \underline{v}(t_i) \quad (2)$$

where

$\underline{x}(t)$ = system state vector at time t (dimension n)

$\underline{F}(t)$ = dynamics matrix at time t (dimension $n \times n$)

$\underline{B}(t)$ = control input matrix at time t (dimension $n \times c$)

$\underline{u}(t)$ = control input vector to the system at time t
(dimension c)

$\underline{G}(t)$ = noise input matrix at time t (dimension $n \times s$)

$\underline{w}(t)$ = input white, Gaussian noise vector (dimension s)

with statistics: $E[\underline{w}(t)] = \underline{0}$

$$E[\underline{w}(t)\underline{w}^T(t+\tau)] = \underline{Q}(t) \delta(\tau)$$

$\underline{z}(t_i)$ = measurement vector at time t_i (dimension m)

$\underline{H}(t_i)$ = measurement matrix at time t_i (dimension $m \times n$)

$\underline{v}(t_i)$ = discrete-time, white, Gaussian noise vector

(dimension m) with statistics $E[\underline{v}(t_i)] = \underline{0}$

$$E[\underline{v}(t_i)\underline{v}^T(t_j)] =$$

$$\underline{R}(t_i) \delta_{ij}$$

In the more general case, where the dynamics and measurement equations are not constrained to be linear, the dynamics and measurement equations become

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t), \underline{u}(t), t] + \underline{G}(t)\underline{w}(t) \quad (3)$$

$$\underline{z}(t_1) = \underline{h}[\underline{x}(t_1), t_1] + \underline{v}(t_1) \quad (4)$$

where

$\underline{f}[\underline{x}(t), \underline{u}(t), t]$ = dynamics equation vector which relates $\underline{x}(t)$ and $\dot{\underline{x}}(t)$ at time t (dimension n)

$\underline{h}[\underline{x}(t_1), t_1]$ = measurement equation vector at time t_1
(dimension m)

In both the linear and nonlinear case, the initial conditions are given by the random Gaussian vector $\underline{x}(t_0)$. Additionally, the dynamics driving noise $\underline{w}(t)$ and the measurement corruption noise $\underline{v}(t)$ are always added linearly, and $\underline{w}(t)$, $\underline{v}(t)$, and $\underline{x}(t_0)$ are assumed uncorrelated. These assumptions are inherent in the derivation of the basic and extended Kalman filter, and will be adhered to throughout this thesis.

2.3 The Basic Kalman Filter

The basic Kalman filter assumes a system with linear dynamics and measurements, given by Equations (1) and (2). Equation (1) expresses the continuous time relationship between $\underline{x}(t)$ and $\dot{\underline{x}}(t)$; this equation can be discretized by using the state transition matrix associated with $\underline{F}(t)$ to give

$$\underline{x}(t_1) = \underline{\phi}(t_1, t_{i-1}) \underline{x}(t_{i-1}) + \int_{t_{i-1}}^{t_1} \underline{\phi}(t_1, \tau) [\underline{B}(\tau) \underline{u}(\tau) + \underline{G}(\tau) \underline{w}(\tau)] d\tau \quad (5)$$

where $\underline{\phi}(t_1, t_{i-1})$ = state transition matrix from time t_{i-1} to time t_1 (dimension $n \times n$)

If we further assume that the control vector $\underline{u}(t)$ is held constant over the sample period, equation (5) becomes

$$\underline{x}(t_1) = \underline{\phi}(t_1, t_{i-1}) \underline{x}(t_{i-1}) + \underline{B}_d(t_{i-1}) \underline{u}(t_{i-1}) + \int_{t_{i-1}}^{t_1} \underline{\phi}(t_1, \tau) \underline{G}(\tau) \underline{w}(\tau) d\tau \quad (6)$$

where

$$\underline{B}_d(t_{i-1}) \triangleq \int_{t_{i-1}}^{t_1} \underline{\phi}(t_1, \tau) \underline{B}(\tau) d\tau$$

The basic Kalman filter propagation equations are approximated by

$$\hat{\underline{x}}(t_1^-) = \underline{\phi}(t_1, t_{i-1}) \hat{\underline{x}}(t_{i-1}^+) + \underline{B}_d(t_{i-1}) \underline{u}(t_{i-1}) \quad (7)$$

$$\begin{aligned} \underline{P}(t_1^-) &= \underline{\phi}(t_1, t_{i-1}) \underline{P}(t_{i-1}^+) \underline{\phi}^T(t_1, t_{i-1}) \\ &+ \int_{t_{i-1}}^{t_1} \underline{\phi}(t_1, \tau) \underline{G}(\tau) \underline{Q}(\tau) \underline{G}^T(\tau) \underline{\phi}^T(t_1, \tau) d\tau \end{aligned} \quad (8)$$

where

$\underline{P}(t_1)$ = the filter-generated state estimate error covariance matrix at time t_1 (dimension $n \times n$)

The $-$ and $+$ superscripts on the time arguments in Equations (7) and (8) and all subsequent equations indicate before and after the measurement is incorporated, respectively.

$\underline{P}(t_1^-)$, for example, represents the error covariance matrix at time t_1 , before the measurement has been incorporated.

The exact form for the propagation equations (no longer assuming a constant $u(t)$ over the sample period) is

$$\dot{\underline{x}}(t/t_{1-1}) = \underline{F}(t)\underline{x}(t/t_{1-1}) + \underline{B}(t)\underline{u}(t) \quad (9)$$

$$\dot{\underline{P}}(t/t_{1-1}) = \underline{F}(t)\underline{P}(t/t_{1-1}) + \underline{P}(t/t_{1-1})\underline{F}^T(t) + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) \quad (10)$$

to be integrated from t_{1-1} to t_1 using the initial conditions

$$\underline{\hat{x}}(t_{1-1}/t_{1-1}) = \underline{\hat{x}}(t_{1-1}^+)$$

$$\underline{P}(t_{1-1}/t_{1-1}) = \underline{P}(t_{1-1}^+)$$

The filter update equations at the next measurement sample time t_1 are

$$\underline{K}(t_1) = \underline{P}(t_1^-)\underline{H}^T(t_1)[\underline{H}(t_1)\underline{P}(t_1^-)\underline{H}^T(t_1) + \underline{R}(t_1)]^{-1} \quad (11)$$

$$\underline{\hat{x}}(t_1^+) = \underline{\hat{x}}(t_1^-) + \underline{K}(t_1)[\underline{Z}(t_1) - \underline{H}(t_1)\underline{\hat{x}}(t_1^-)] \quad (12)$$

$$\underline{P}(t_1^+) = \underline{P}(t_1^-) - \underline{K}(t_1)\underline{H}(t_1)\underline{P}(t_1^-) \quad (13)$$

where

$\underline{K}(t_1)$ = filter gain matrix at time t_1 (dimension $n \times m$)

It can be shown that the solution to the linear stochastic differential equation of Equation (6) is a Gauss-Markov process, and that $\underline{x}(t)$ for any fixed time t is a Gaussian random variable (5:164-168). It can be further shown that the estimate generated by the Kalman filter is, in the case of linear dynamics and measurement equations, the optimum estimate in the following respects;

- 1) $\hat{\underline{x}}(t_1^+)$ is the mean, mode, and median of the probability density function of the state vector, conditioned on the entire measurement history (5:206-226).
- 2) $\hat{\underline{x}}(t_1^+)$ is the minimum mean square error estimate (5:232).
- 3) $\hat{\underline{x}}(t_1^+)$ is the maximum a posteriori estimate (5:234).

2.4 The Extended Kalman Filter (26:42-59)

The extended Kalman filter is an extension of the Kalman filter (as the name might imply) to the case where either $\underline{f}[\underline{x}(t), \underline{u}(t), t]$ in Equation (3), or $\underline{h}[\underline{x}(t_1), t_1]$ in Equation (4), or both, are nonlinear functions. The EKF repeatedly re-linearizes the dynamics and measurement equations by assuming that perturbations of the state vector $\underline{x}(t)$ from a repeatedly redefined nominal are small over the sample period. Thus, while the linear Kalman filter guarantees an optimal estimate, the EKF estimate is "optimal" only to the extent that the small perturbation assumption is true.

The propagation equations for the EKF are

$$\dot{\hat{\underline{x}}}(t/t_{1-1}) = \underline{f}[\hat{\underline{x}}(t/t_{1-1}), \underline{u}(t), t] \quad (14)$$

$$\begin{aligned} \dot{\underline{P}}(t/t_{1-1}) = & \underline{F}[t; \hat{\underline{x}}(t/t_{1-1})] \underline{P}(t/t_{1-1}) + \\ & \underline{P}(t/t_{1-1}) \underline{F}^T[t; \hat{\underline{x}}(t/t_{1-1})] + \underline{G}(t) \underline{Q}(t) \underline{G}^T(t) \end{aligned} \quad (15)$$

to be integrated from t_{1-1} to t_1 using the initial conditions

$$\hat{\underline{x}}(t_{1-1}/t_{1-1}) = \hat{\underline{x}}(t_{1-1}^+)$$

$$\underline{P}(t_{1-1}/t_{1-1}) = \underline{P}(t_{1-1}^+)$$

where

$$\underline{F}[t; \hat{\underline{x}}(t/t_{1-1})] = \left. \frac{\partial \underline{f}[\underline{x}, \underline{u}(t), t]}{\partial \underline{x}} \right|_{\underline{x} = \hat{\underline{x}}(t/t_{1-1})}$$

The filter update equations are

$$\begin{aligned} \underline{K}(t_1) = & \underline{P}(t_1^-) \underline{H}^T[t_1; \hat{\underline{x}}(t_1^-)] \{ \underline{H}[t_1; \hat{\underline{x}}(t_1^-)] \underline{P}(t_1^-) \underline{H}^T[t_1; \hat{\underline{x}}(t_1^-)] \\ & + \underline{R}(t_1) \} \end{aligned} \quad (16)$$

$$\hat{\underline{x}}(t_1^+) = \hat{\underline{x}}(t_1^-) + \underline{K}(t_1) [\underline{Z}(t_1) - \underline{h}[\hat{\underline{x}}(t_1^-), \underline{u}(t_1), t_1]] \quad (17)$$

$$\underline{P}(t_1^+) = \underline{P}(t_1^-) - \underline{K}(t_1) \underline{H}[t_1; \hat{\underline{x}}(t_1^-)] \underline{P}(t_1^-) \quad (18)$$

where

$$H[t_1; \hat{x}(t_1^-)] = \frac{\partial h[\underline{x}, \underline{u}(t_1), t_1]}{\partial \underline{x}} \bigg|_{\underline{x} = \hat{\underline{x}}(t_1^-)}$$

As in the basic Kalman filter, the initial conditions are modelled as a random vector \underline{x} with mean $\hat{\underline{x}}(t_0)$ and covariance $\underline{P}(t_0)$.

Although these equations appear similar to the propagation and update equations given in Section 2.3 for the basic Kalman filter (Equations (9) - (13)), there is a fundamental difference. The error covariance matrix in the basic Kalman filter is independent of the measurements and the estimated values of the state vector, and can thus be precomputed and analyzed without knowledge of the actual measurements. The error covariance matrix of the EKF, on the other hand, is dependent on the state estimate time history through the recomputed nominal. This point is significant in determining a method for evaluating the Kalman filter algorithm, and will be discussed further in Chapter 4, where a Monte Carlo analysis will be described.

CHAPTER III

ALGORITHM DEVELOPMENT

3.1 Introduction

This chapter will discuss the development of the six algorithms evaluated in this thesis. The purpose of these algorithms is to obtain the best estimate of the range of a moving, airborne target from a moving, airborne tracker, where the tracker is measuring only azimuth, elevation, azimuth rate, and elevation rate of the line of sight to the target. Each method attempts to arrive at a solution by finding a target range and velocity which is consistent with the measured angles and angle rates coming from the tracker, a problem which is compounded by noise in the tracker measurements and by limited adequacy of proposed mathematical models. All algorithms are based on extended Kalman filters; they differ in their state vector representation and target motion model assumptions.

It should be noted that some of algorithms use nonlinear dynamics equations and linear measurement equations, while others use linear dynamics equations and nonlinear measurement equations. From the discussion on Kalman filters, one can see that the implementation of the former algorithm would be more burdensome computationally than the implementation of the latter, since some kind of online integration routine would have to be used for propagation in the case of nonlinear dynamics. Therefore, if equal performance were obtained from these two types of algorithms, the second would be the better choice from the standpoint of computational loading.

In an attempt to minimize, if not eliminate, the confusion caused by presenting six algorithms using different state vectors and different coordinate systems, the second section of this chapter is devoted to the definition of reference frames and notation conventions which will remain in effect throughout the rest of the thesis. The subsequent sections of this chapter will present in detail the six algorithms to be evaluated.

3.2 Reference Frames and Notation

There are four different reference frames used in the ranging algorithms to be evaluated; three frames using Cartesian coordinates, differing only in origin and orientation, and the fourth frame using spherical coordinates. Because the state vector of the six algorithms all contain states expressed in two different coordinate systems, it is important to understand these reference frames before proceeding to the algorithm descriptions. The four reference frames will be referred to as inertial north-east-down (NED), relative NED, line of sight (LOS), and spherical.

3.2.1 Reference Frames. The inertial NED frame is a Cartesian coordinate frame which is oriented so that the X, Y, and Z axes point north, east, and down, respectively. For this thesis, north is defined as true north, down is defined as parallel to local gravity, and east is perpendicular to the other two axes to form a right-hand orthogonal coordinate system. This "reference frame" is really a series of frames, each of which is fixed with respect to the earth and oriented

north-east-down. At each measurement update time, the "inertial NED" frame is chosen as that frame in the series whose origin is at the (attacking) fighter aircraft. While these frames are not truly inertial, as they are fixed to a translating, rotating earth, they can be considered inertial over the relatively short times and distances involved in this thesis.

The relative NED frame differs from the inertial NED frame in that the relative NED frame origin is always located at the fighter aircraft and translates with the motion of the fighter. The axes of the inertial NED frame and the relative NED frame are considered to be aligned at the sequence of points where the "inertial NED" frame is defined.

The statement that the orientation of the axes in both the inertial NED frame and the relative NED frame are always north, east, and down, implies that these frames must rotate as the fighter aircraft moves across the curved surface of the earth. Thus at the instant of measurement update, the origin of the inertial NED frame becomes fixed with respect to the earth, and although the axes of the relative and inertial frames are aligned, the relative frame has an instantaneous rotation rate with respect to the inertial frame about an axis perpendicular to the fighter aircraft's velocity vector (assuming a spherical earth). This rotation rate is relatively small, but will be included in the elevation rate measurement equations of the ranging algorithms.

The third reference frame to be described is the LOS

frame. This is again a Cartesian coordinate system, with the axes oriented so that the X axis points along the line of sight from the fighter to the target, the Y axis is perpendicular to the X axis and always in the horizontal plane, and the Z axis is perpendicular to the other two and generally "down" to form a right-hand orthogonal coordinate system (Figure 3-1). As in the inertial NED frame, the LOS frame can be thought of as a series of frames, each fixed with respect to the earth. At measurement update time, the LOS frame is chosen as that earth-fixed frame whose axes are aligned as described above and whose origin is at the fighter aircraft. The rotation of this frame is thus ignored, as in the inertial NED frame. This is less valid here, however, because the rotation of the axes of the LOS frame is caused by the relative movement of the target and the fighter, not just the curvature of the earth's surface. The errors caused by assuming away the conceivably large rotation rate of the LOS frame will be made graphically apparent in Chapter 5, in which the algorithm performance is discussed.

The last frame of reference considered here is in spherical coordinates, with the origin located at the fighter aircraft and translating with it. Positions in this reference frame are expressed in terms of a distance (R) at some azimuth (θ) and elevation (γ). Azimuth is defined as the angle in the horizontal plane between the north axis and the projection of the position vector onto the horizontal plane, measured positive clockwise from north; elevation is the angle

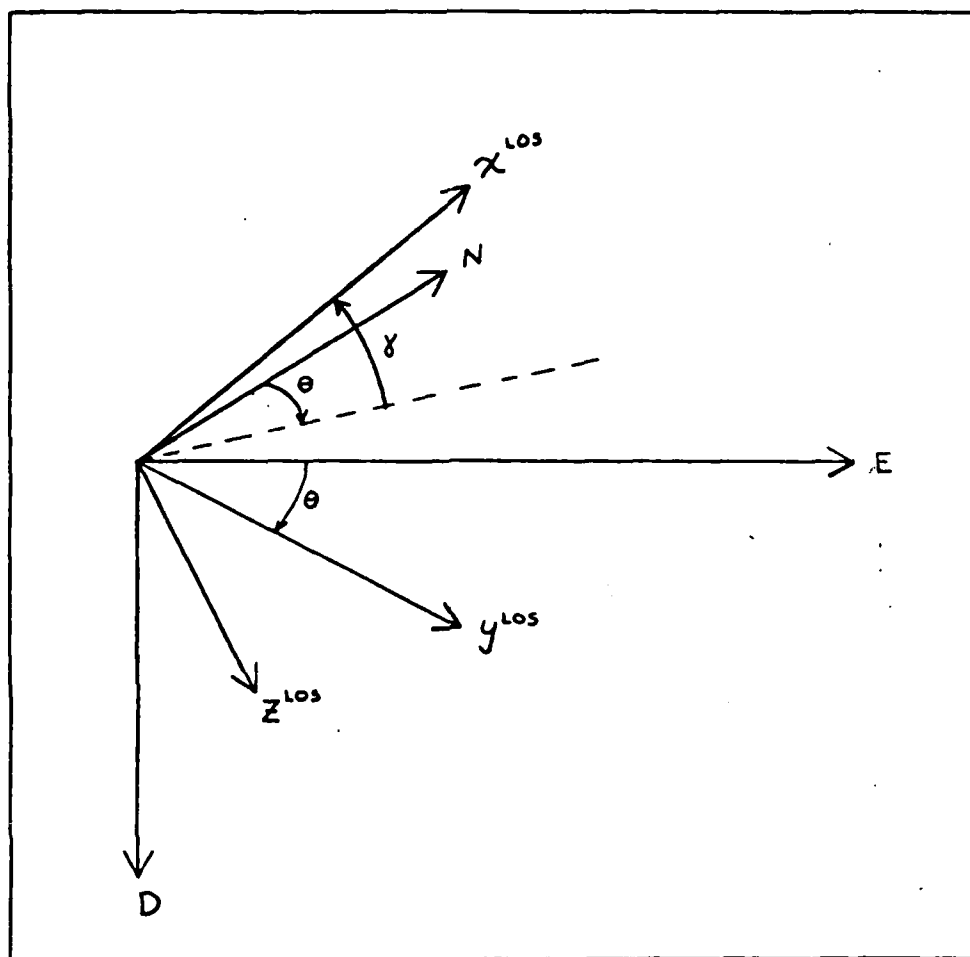


Figure 3-1. Line of Sight Reference Frame

Note: The N, E, and y^{LOS} axes are in the horizontal (local level) plane. The x^{LOS} , z^{LOS} , and D axes are in the same vertical plane, which is offset by azimuth, θ , from the N axis.

between the horizontal plane and the position vector, measured positive up (Figure 3-2).

3.2.2 Notation. This section establishes notation and subscript convention which will be used throughout the rest of the thesis. All angles and angle rates will be expressed in microradians and microradians per second, respectively, unless otherwise noted. The coordinate frame in which the quantities are expressed is always the frame with respect to which the time derivative is taken; i.e., velocity coordinated in the line of sight frame implies the time derivative of position with respect to the line of sight frame.

Position in spherical coordinates will be represented as R^S , θ , and γ , the superscript S on R denoting the spherical frame. The time derivatives of these quantities will be written as \dot{R}^S , $\dot{\theta}$, and $\dot{\gamma}$.

Position, velocity, and acceleration in the Cartesian coordinate systems will be written as x, y, z ; V_x, V_y, V_z ; and A_x, A_y, A_z . These will be superscripted with I, REL, or LOS, to indicate reference to the inertial NED frame, relative NED frame, or line of sight frame, respectively. Further, velocities and accelerations will have a second subscript of F or T to indicate fighter or target motion.

For example, V_{XF}^{LOS} indicates fighter velocity with respect to the LOS frame and in the x direction of the LOS frame; A_{YT}^I indicates target acceleration with respect to the inertial NED frame and in the y(east) direction of the inertial NED frame.

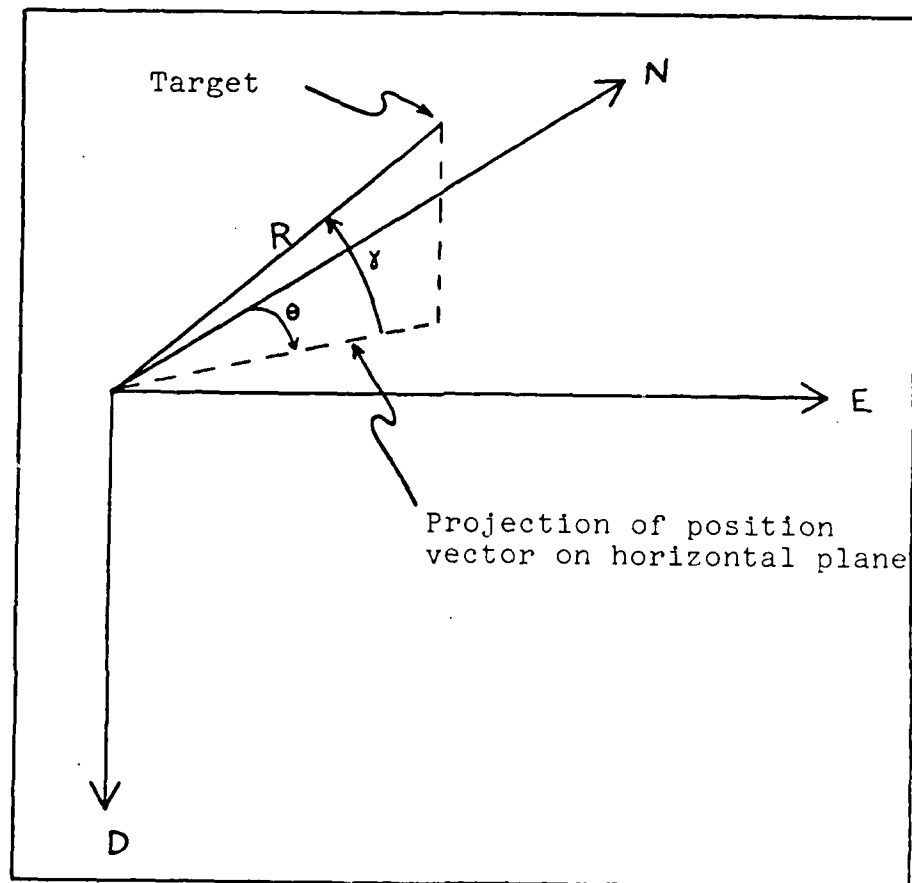


Figure 3-2. Spherical Frame

Table 3-1 provides a summary of the notation. Position expressions in the inertial NED and the LOS frames, and acceleration in the relative NED frame, are omitted, since they are not used in any of the evaluated algorithms.

Table 3-1 Notation Summary

Reference Frame	Position	Velocity (target)	Acceleration (fighter)
Inertial NED	N/A	V_{XT}^I	A_{XF}^I
	N/A	V_{YT}^I	A_{YF}^I
	N/A	V_{ZT}^I	A_{ZF}^I
Relative NED	X^{REL}	V_{XT}^{REL}	N/A
	Y^{REL}	V_{YT}^{REL}	N/A
	Z^{REL}	V_{ZT}^{REL}	N/A
Line of Sight	N/A	V_{XT}^{LOS}	A_{XF}^{LOS}
	N/A	V_{YT}^{LOS}	A_{YF}^{LOS}
	N/A	V_{ZT}^{LOS}	A_{ZF}^{LOS}
Spherical	R^S	\dot{R}^S	\ddot{R}^S
	θ	$\dot{\theta}$	$\ddot{\theta}$
	γ	$\dot{\gamma}$	$\ddot{\gamma}$

3.3 Algorithm Description

3.3.1 Overview. Section 3.3 describes in detail the six algorithms which are tested in this thesis. The algorithms are numbered from one to six in the order in which they are presented, which is generally in order of decreasing complexity. Before the detailed descriptions, however, a brief overview of the physical reasoning behind the target models used is in order.

The first two algorithms model target acceleration as three independent acceleration processes, each the output of a first order lag driver by white, Gaussian noise. Physically, this means that target acceleration in any direction is time-correlated (the target cannot accelerate instantaneously) and that the target acceleration in any direction is independent of the accelerations in the other two directions. In some scenarios, such as a close-in gun tracking solution, for example, it may be possible to specify a target model with accelerations which are independent from channel to channel in the LOS frame but each with a different λ_0 value. If this were done, these accelerations would not be independent from channel to channel in the inertial NED frame if the inertial NED frame and LOS frame were not aligned. In this thesis, the strengths of the white driving noise and hence the λ_0 values of the accelerations of the three channels are assumed equal, so that the target models are equivalent in the inertial NED and LOS frames.

This is a very flexible target model, and can be made to match a wide range of potential targets by appropriate

choices for the time constant of the lag, τ , and the strength of the dynamics driving noise, $Q(t)$ (see Section 3.4).

In fact, because the accelerations are not correlated with target velocity (as opposed to an actual aircraft, which has a greater acceleration ability in a direction perpendicular to aircraft velocity), the target in this model has more freedom of movement than an actual target would have.

The target model used in algorithm four is an attempt to limit to some degree the target's freedom of movement. In this model, the target is assumed to have a relatively constant total velocity magnitude (very little acceleration along the velocity vector), and to have a maneuver capability only of turns in the horizontal plane. This is physically reasonable, as aircraft structural limits and energy management considerations, as well as some tactical considerations, tend to limit aircraft motion to horizontal maneuvers. Heading rate in this algorithm is modelled as Brownian motion; that is, the time derivative of heading rate is modelled as white, zero-mean, Gaussian noise:

$$\frac{d}{dt} (\dot{\alpha}) = 0 + w(t).$$

Physically, this is saying that the target has a basically constant heading rate (with wideband noise added to model the inadequacy of a completely constant rate assumption), or is essentially in a constant airspeed, constant g horizontal turn. While heading rate could have been modelled in the same way as the accelerations in algorithms one and two, it

was felt that, considering the relatively small heading rates really involved and the speed with which they could be changed, this would needlessly complicate the algorithm. Section 3.4 explains the method for finding the correct strength for the white noise.

Algorithms three (so numbered because of its similarity to algorithm two), five, and six limit target motion still further, with velocity modelled as Brownian motion in the same manner as heading rate above, i.e., $\dot{\underline{V}}(t) = 0 + \underline{w}(t)$. Again, time correlation was not modelled, as the velocity is expected to be basically constant, and the addition of a first order lag would needlessly complicate the algorithm. The target in this case is limited to flying in basically a straight line, with small excursions handled by the noise. The determination of appropriate values for $\underline{Q}(t)$, the strength of the noise, is discussed in Section 3.4.

In all algorithms, the fighter aircraft motion is assumed to be known perfectly from its inertial navigation system. While this is not a completely accurate representation, it is a reasonable assumption considering the short times and distances involved in this problem, and the relative precision of the INS data and data from the tracking sensor. If this assumption were used in actual algorithm implementation, consideration may be given to adding a small amount of pseudonoise to "tell the filter" that these measurements are not really precise.

3.3.2 Algorithm One. The first algorithm employs a state vector in which position and velocity are expressed in the spherical coordinate frame, while target acceleration is expressed in the LOS frame. This allows target acceleration to be modelled with respect to the earth, since the LOS frame as defined here is earth-fixed, while position and velocity in the spherical frame are relative to the fighter aircraft. This leads to a state vector of

$$\begin{bmatrix} R^S \\ \dot{R}^S \\ A_{XT}^{LOS} \\ \theta \\ \dot{\theta} \\ A_{YT}^{LOS} \\ \gamma \\ \dot{\gamma} \\ A_{ZT}^{LOS} \end{bmatrix}$$

where the elements are as defined in Section 3.2.2.

In this approach, as previously mentioned, we will assume that the Cartesian target acceleration is well modelled as three independent acceleration processes, each the output of a first order lag driven by white, Gaussian noise. In order to write out the dynamics equations, however, we must find a way to relate \ddot{R}^S , $\ddot{\theta}$, and $\ddot{\gamma}$ to the other elements of the state vector. We will do this by defining a vector \underline{R} to be the

position vector of the target aircraft relative to the fighter aircraft, and differentiating twice with respect to time as seen in the LOS frame. (This development parallels one by R.A.K. Mitchell, Ref 7, pp 19-21).

Letting $\underline{\omega}$ be the angular velocity of the spherical frame with respect to the LOS frame, which, again, is considered inertial, we can use the Theorem of Coriolis to write

$$\left. \frac{d^2 \underline{R}}{dt^2} \right|_{\text{LOS}} = \left. \frac{d^2 \underline{R}}{dt^2} \right|_S + 2 \underline{\omega} \times \left. \frac{d \underline{R}}{dt} \right|_S + \left. \frac{d \underline{\omega}}{dt} \right|_S \times \underline{R} + \underline{\omega} \times (\underline{\omega} \times \underline{R}) \quad (19)$$

where the vertical line denotes the frame in which the derivative is taken (7:19).

The left side of Equation (19), when written in LOS coordinates, is the relative acceleration of the target with respect to the fighter, or

$$\left. \frac{d^2 \underline{R}^{\text{LOS}}}{dt^2} \right|_{\text{LOS}} = \begin{bmatrix} A_{XT}^{\text{LOS}} \\ A_{YT}^{\text{LOS}} \\ A_{ZT}^{\text{LOS}} \end{bmatrix} - \begin{bmatrix} A_{XF}^{\text{LOS}} \\ A_{YF}^{\text{LOS}} \\ A_{ZF}^{\text{LOS}} \end{bmatrix} \quad (20)$$

If expressed in the LOS frame in terms of azimuth and elevation rate, $\underline{\omega}$ can be written as (See Figure 3.3):

$$\underline{\omega}^{\text{LOS}} = \begin{bmatrix} -\dot{\theta} \sin \gamma \\ \dot{\gamma} \\ \dot{\theta} \cos \gamma \end{bmatrix} \quad (21)$$

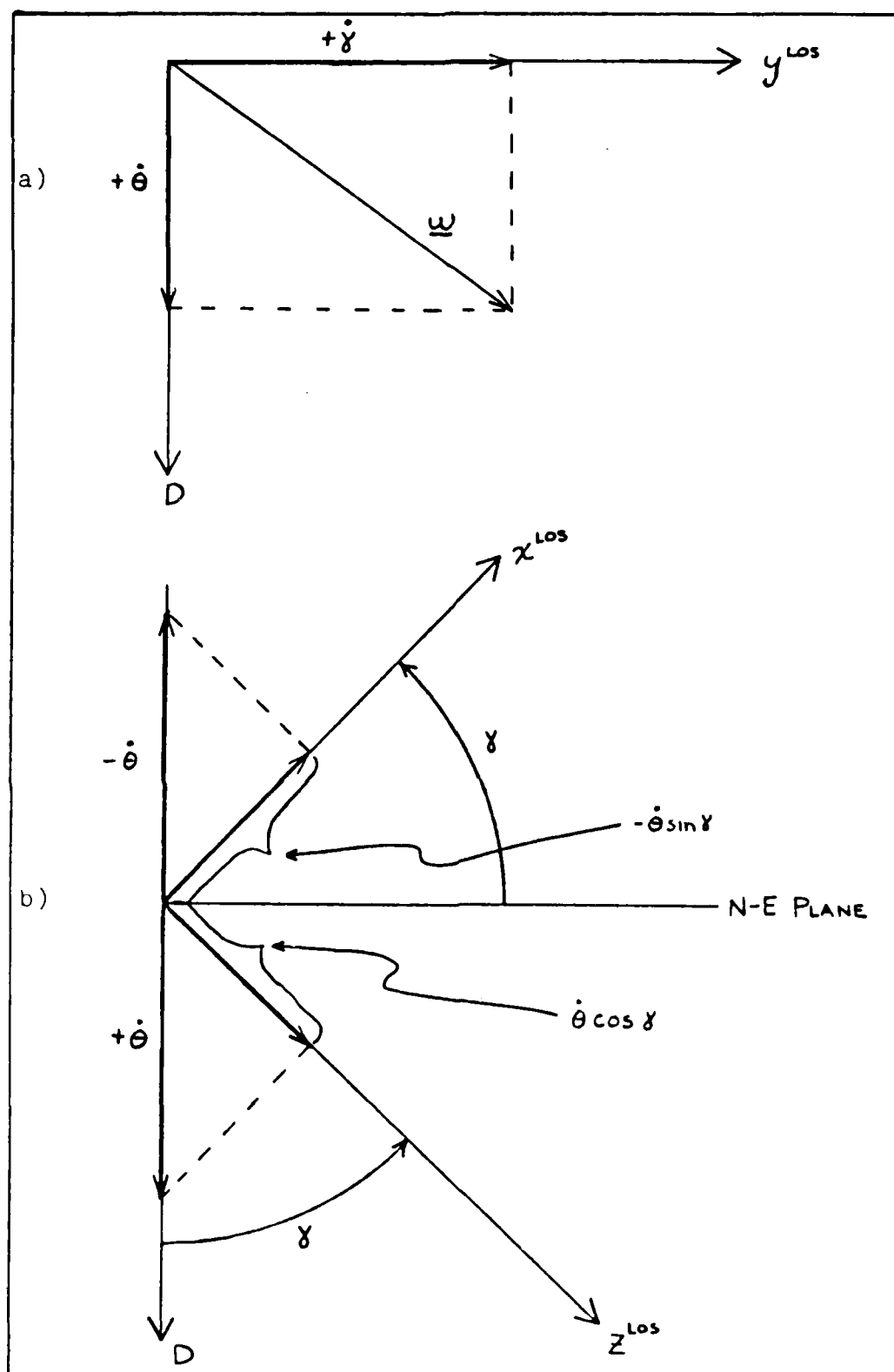


Figure 3-3. Rotation Rate in the LOS Frame

a) In the y^{LOS} - D plane

b) In the x^{LOS} - D plane

Additionally, because the position vector \underline{R} lies completely along the X axis in the LOS frame, and R^S in spherical coordinates lies along the same line, we can write

$$\left. \frac{d^2 \underline{R}^{LOS}}{dt^2} \right|_S = \begin{bmatrix} \ddot{R}^S \\ 0 \\ 0 \end{bmatrix} \quad (22a)$$

$$\left. \frac{d \underline{R}^{LOS}}{dt} \right|_S = \begin{bmatrix} \dot{R}^S \\ 0 \\ 0 \end{bmatrix} \quad (22b)$$

$$\underline{R}^{LOS} = \begin{bmatrix} R^S \\ 0 \\ 0 \end{bmatrix} \quad (22c)$$

Substituting into Equation (19) and taking the cross products yields

$$\begin{bmatrix} A_{XT}^{LOS} - A_{XF}^{LOS} \\ A_{YT}^{LOS} - A_{YF}^{LOS} \\ A_{ZT}^{LOS} - A_{ZF}^{LOS} \end{bmatrix} = \begin{bmatrix} \ddot{R}^S - R^S \dot{\gamma}^2 - R^S \cos^2 \gamma \dot{\theta}^2 \\ R^S (\ddot{\theta} \cos \gamma - 2 \dot{\theta} \dot{\gamma} \sin \gamma) + 2 \dot{R}^S \cos \gamma \dot{\theta} \\ -R^S (\ddot{\gamma} + \cos \gamma \sin \gamma \dot{\theta}^2) - 2 \dot{R}^S \dot{\gamma} \end{bmatrix} \quad (23)$$

Manipulating to isolate \ddot{R}^S , $\ddot{\theta}$, and $\ddot{\gamma}$, and expressing the angles in microradians gives

$$\ddot{R}^S = A_{XT}^{LOS} - A_{XF}^{LOS} + R^S \dot{\gamma}^2 \times 10^{-12} + R^S \dot{\theta}^2 \cos^2 \gamma \times 10^{-12} \quad (24a)$$

$$\ddot{\theta} = \frac{(A_{YT}^{LOS} - A_{YF}^{LOS}) \times 10^6}{R^S \cos \gamma} - \frac{2R^S \dot{\theta}}{R^S} + 2\dot{\gamma} \dot{\theta} \tan \gamma \times 10^{-6} \quad (24b)$$

$$\ddot{\gamma} = - \frac{(A_{ZT}^{LOS} - A_{ZF}^{LOS}) \times 10^6}{R^S} - \frac{2R^S \dot{\gamma}}{R^S} - \frac{\dot{\theta}^2 \sin 2\gamma \times 10^{-6}}{2} \quad (24c)$$

The above equations form the basis for the dynamics equations for algorithm one. Before going on, however, the discrepancy of the "inertial" LOS frame which remains aligned with a moving position vector should again be noted. The LOS frame is not truly inertial; it is assumed that it is approximately inertial for relatively small elevation and azimuth rates, such as would occur if the fighter aircraft were on a near collision course with the target.

The dynamics equation vector to be used in Equation (3) for this algorithm is given by

$$\underline{f}[\underline{x}(t), \underline{u}(t), t] = \begin{bmatrix} \dot{R}^S \\ A_{XT}^{LOS} - A_{XF}^{LOS} + (R^S \dot{\gamma}^2 + R^S \dot{\theta}^2 \cos^2 \gamma) \times 10^{-12} \\ -\frac{1}{\tau} A_{XT}^{LOS} \\ \dot{\theta} \\ (A_{YT}^{LOS} - A_{YF}^{LOS}) \times 10^6 / (R^S \cos \gamma) - 2R^S \dot{\theta} / R^S + 2\dot{\gamma} \dot{\theta} \tan \gamma \times 10^{-6} \\ -\frac{1}{\tau} A_{YT}^{LOS} \\ \dot{\gamma} \\ (A_{ZT}^{LOS} - A_{ZF}^{LOS}) \times 10^6 / R^S - 2R^S \dot{\gamma} / R^S - \dot{\theta}^2 \sin 2\gamma \times 10^6 / 2 \\ -\frac{1}{\tau} A_{ZT}^{LOS} \end{bmatrix} \quad (25)$$

Examination of Equation (25) reveals that the angle θ is not used in any of the equations in the vector, and γ is used only as the argument of trigonometric functions. It is reasonable to assume that, given inertial angle measurement accuracy of 1 milliradian or better, we could delete the angles θ and γ from the state vector and simply use the measured angle γ in the propagation equations for the other states. Again, since θ does not appear in any of the propagation equations, this can obviously be deleted with no effect. The introduction of the noisy elevation measurement directly into the equations, however, will cause some error in the target state, and may require the addition of pseudonoise in those channels in which γ is used (in the propagation of \dot{R}^S , $\dot{\theta}$, and $\dot{\gamma}$). The new state vector is

$$\begin{bmatrix} R^S \\ \dot{R}^S \\ A_{XT}^{LOS} \\ \dot{\theta} \\ A_{YT}^{LOS} \\ \dot{\gamma} \\ A_{CT}^{LOS} \end{bmatrix}$$

The dynamics equation, Equation (3), written in vector form, becomes

$$\frac{d}{dt} \begin{bmatrix} R^S \\ \dot{R}^S \\ A_{XT}^{LOS} \\ \dot{\theta} \\ A_{YT}^{LOS} \\ \dot{\gamma} \\ A_{ZT}^{LOS} \end{bmatrix} = \begin{bmatrix} \dot{R}^S \\ A_{XT}^{LOS} - A_{XF}^{LOS} + (R^S \dot{\gamma}^2 + R^S \dot{\theta}^2 \cos^2 \gamma) \times 10^{-12} \\ -\frac{1}{\tau} A_{XT}^{LOS} \\ (A_{YT}^{LOS} - A_{YF}^{LOS}) \times 10^6 / (R^S \cos \gamma) - 2\dot{R}^S + 2\dot{\gamma}\dot{\theta} \tan \gamma \times 10^{-6} \\ -\frac{1}{\tau} A_{YT}^{LOS} \\ (A_{ZF}^{LOS} - A_{ZT}^{LOS}) \times 10^6 / R^S - 2\dot{R}^S \dot{\gamma} / R^S - \dot{\theta}^2 \sin 2\gamma \times 10^6 / 2 \\ -\frac{1}{\tau} A_{ZT}^{LOS} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ W_1 \\ 0 \\ W_2 \\ 0 \\ W_3 \end{bmatrix} \quad (26)$$

Since the measurements coming from the sensor are inertial angle rates, we are actually measuring states four and six of the state vector directly. Thus the measurement equation, Equation (4), becomes

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \dot{\gamma} + \omega_C \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (27a)$$

where ω_C is the rotation rate of the spherical reference frame with respect to the LOS frame, caused by fighter velocity over the curved surface of the earth, as discussed in Section 3.2.1. ω_C , in microradians/second, is defined by

$$\omega_C = (V_{XF}^I \cos \theta + V_{YF}^I \sin \theta) \times 10^6 / R_e \quad (27b)$$

where R_e is the radius of the (assumed spherical) earth. Note that when spherical coordinates are used in the state vector to express position and velocity, the dynamics equations are nonlinear, while the measurement equations are linear. We will see that the opposite is true when the state vector is expressed entirely in Cartesian coordinate frames. As mentioned previously, given equal performance of these two algorithms, computational loading considerations would lead toward the choice of linear dynamics and nonlinear measurements.

The EKF propagation and update equations are given by Equations (14)-(18), with Equation (26) and (27) describing the appropriate dynamics vector and measurement vector, respectively.

The \underline{G} , \underline{F} , and \underline{H} matrices needed for implementation are given in Appendix A; the selection of appropriate values for the \underline{Q} and \underline{R} matrices and the time constant τ are discussed in Section 3.4.

3.3.3 Algorithm Two. The state vector used in the second algorithm is expressed entirely in Cartesian coordinate frames: target position and velocity in the relative NED frame, and target acceleration in the inertial NED frame. As in algorithm one, target acceleration is with respect to an earth-fixed reference frame, hopefully leading to a more accurate target model.

The state vector is

$$\begin{bmatrix} x^{REL} \\ y^{REL} \\ z^{REL} \\ v_{XT}^{REL} \\ v_{YT}^{REL} \\ v_{ZT}^{REL} \\ A_{XT}^I \\ A_{YT}^I \\ A_{ZT}^I \end{bmatrix}$$

Target acceleration is modelled again as three independent acceleration processes, each the output of a first order lag driven by white, Gaussian noise. The dynamics equations for this algorithm are linear and relatively simple, and, written in vector form, are

$$\frac{d}{dt} \begin{bmatrix} x^{REL} \\ y^{REL} \\ z^{REL} \\ v_{XT}^{REL} \\ v_{YT}^{REL} \\ v_{ZT}^{REL} \\ A_{XT}^I \\ A_{YT}^I \\ A_{ZT}^I \end{bmatrix} = \begin{bmatrix} v_{XT}^{REL} \\ v_{YT}^{REL} \\ v_{ZT}^{REL} \\ A_{XT}^I - A_{XP}^I \\ A_{YT}^I - A_{YP}^I \\ A_{ZT}^I - A_{ZP}^I \\ -\frac{1}{T} A_{XT}^I \\ -\frac{1}{T} A_{YT}^I \\ -\frac{1}{T} A_{ZT}^I \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

(28)

It should be pointed out that in actual implementation, pseudonoise could be added to the velocity channels because the fighter accelerations are not really perfectly known.

The measurement equations for this algorithm are somewhat more complex. Figure 3-4 shows the relationship between the relative NED frame and the position vector to the target. The angles θ and γ , and the angle rates $\dot{\theta}$ and $\dot{\gamma}$, are the measurements which are input to the filter algorithm from the tracker. From Figure 3-4 we can see that, in microradians,

$$\theta = (\tan^{-1} \frac{y^{REL}}{x^{REL}}) \times 10^6 \quad (29)$$

and

$$\gamma = (\tan^{-1} \frac{-z^{REL}}{\sqrt{x^{REL^2} + y^{REL^2}}}) \times 10^6 \quad (30)$$

By either calculating the time derivatives of Equation (29) and (30), or by finding the velocities required to cause angle rates of $\dot{\theta}$ and $\dot{\gamma}$ from Figure 3-3, we can see that

$$\dot{\theta} = \frac{x^{REL} v_{YT}^{REL} - y^{REL} v_{XT}^{REL}}{x^{REL^2} + y^{REL^2}} \times 10^6 \quad (31)$$

and

$$\dot{\gamma} = \left[\frac{v_{XT}^{REL} x^{REL} z^{REL} + v_{YT}^{REL} y^{REL} z^{REL} - v_{ZT}^{REL} (x^{REL^2} + y^{REL^2})}{\sqrt{x^{REL^2} + y^{REL^2}}} \right] \times 10^6 \quad (32)$$

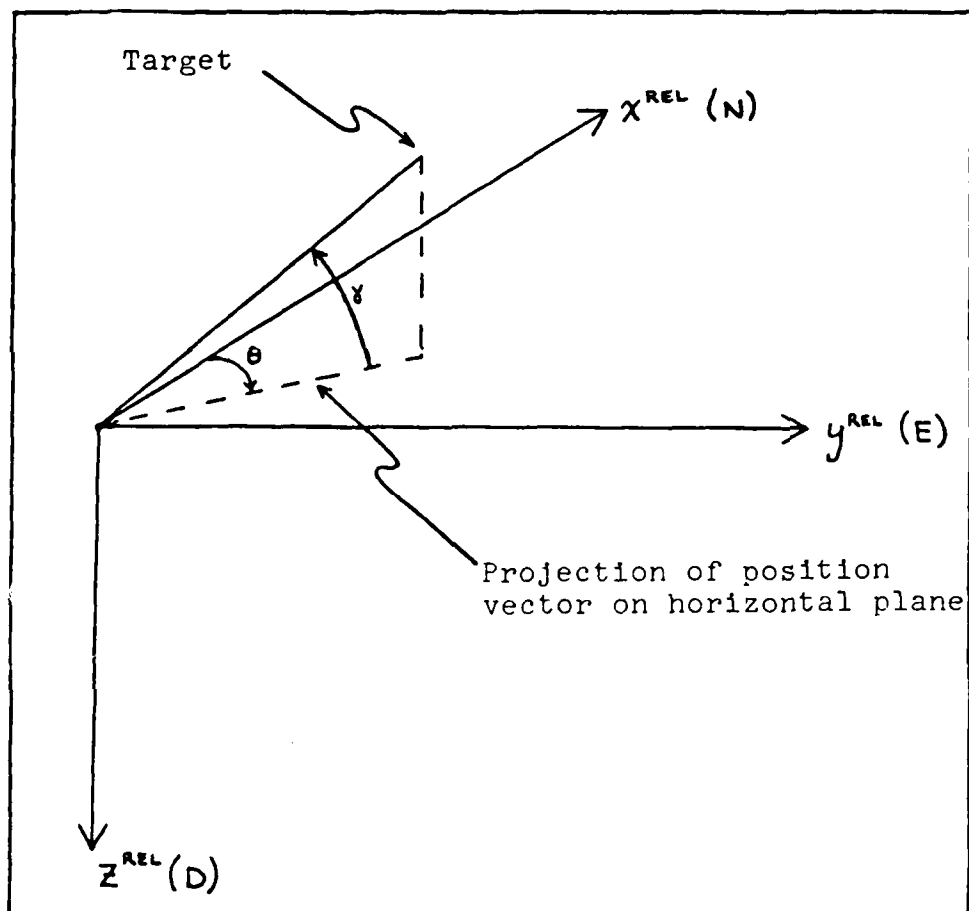


Figure 3-4. Target Position in the Relative NEF Frame

again expressed in microradians/second.

Thus, in vector form, the measurement equations are

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{y^{REL}}{x^{REL}}\right) \times 10^6 \\ \tan^{-1}\left(\frac{-z^{REL}}{\sqrt{x^{REL^2} + y^{REL^2}}}\right) \times 10^6 \\ \frac{x^{REL}v_{YT}^{REL} - y^{REL}v_{XT}^{REL}}{x^{REL^2} + y^{REL^2}} \times 10^6 \\ \frac{v_{XT}^{REL}x^{REL}z^{REL} + v_{YT}^{REL}y^{REL}z^{REL} - v_{ZT}^{REL}(x^{REL^2} + y^{REL^2})}{\sqrt{x^{REL^2} + y^{REL^2}}(x^{REL^2} + y^{REL^2} + z^{REL^2})} + 10^6 + \omega_C \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (33)$$

where ω_C is the correction term for fighter movement over the earth, as discussed in Section 3.2.2, Equation (27b).

In order to estimate the target position in the relative NED frame, the filter must be given the angle measurements as well as the angle rates. This makes the measurement vector a vector of dimension four rather than two, as in algorithm one.

In implementing this algorithm, an effort was made to minimize the errors caused by linearizing the inverse tangent function in the first two measurements by adding bias correction terms to the residual calculations (6:224). These terms

were added in the angle measurement residual calculations and not in the angle rate measurement residual calculations of this and other algorithms because it was believed that the degree of nonlinearity of the inverse tangent function warranted their addition, while the other measurement equations did not. The bias correction terms appear in the EKF state vector update equation, and change Equation (17) to

$$\hat{\underline{X}}(t_1^+) = \hat{\underline{X}}(t_1^-) + \underline{K}(t_1)[\underline{Z}(t_1) - h[\hat{\underline{X}}(t_1^-), \underline{u}(t_1), t_1] - \hat{\underline{b}}_m(t_1^-)] \quad (34)$$

where $\hat{\underline{b}}_m(t_1^-)$ is a vector of dimension m whose k^{TH} component is given as

$$\hat{b}_{mk}(t_1^-) = \frac{1}{2} \text{tr} \left[\frac{\partial^2 h_k \hat{\underline{X}}(t_1^-), \underline{u}(t_1), t_1}{\partial \underline{X}^2} \underline{P}(t_1^-) \right] \quad (35)$$

This has the effect of correcting for nonlinearities in $h[\hat{\underline{X}}(t_1^-), \underline{u}(t_1), t_1]$ by bringing in second order terms without going to a fully developed second order filter.

The remaining matrices required for implementation are given in Appendix B; the selection of appropriate values for $\underline{Q}(t)$, $\underline{R}(t_1)$, and τ is covered in Section 3.4

It should be noted that, because range is a nonlinear function of the estimated states \hat{x}^{REL} , \hat{y}^{REL} , and \hat{z}^{REL} , we

cannot simply say that $\hat{R} = \sqrt{\hat{x}^{REL^2} + \hat{y}^{REL^2} + \hat{z}^{REL^2}}$. Instead, the

equation $R = \sqrt{X^{REL^2} + Y^{REL^2} + Z^{REL^2}}$ must be expanded is a Taylor series about the estimated values of X^{REL} , Y^{REL} , and Z^{REL} , and then the expected value taken of both sides.

Following this procedure, it was found that calculating range by taking the square root of the sum of the square of

the Cartesian estimates, $\hat{R} = \sqrt{\hat{X}^{REL^2} + \hat{Y}^{REL^2} + \hat{Z}^{REL^2}}$, differed from the series expansion range estimate by only approximately 0.01 feet in all calculations throughout the test run. The simpler calculation of the range estimate is therefore used in both this algorithm and the following algorithm.

3.3.4 Algorithm Three. Algorithm three is basically a simplified form of algorithm two, in which target acceleration is modelled as three independent zero mean, white, Gaussian noises, rather than three first-order Gauss Markov processes, as in algorithm two. The state vector is similar to that of the preceeding algorithm, with acceleration deleted and target velocities expressed in inertial NED coordinates. The state vector is

$$\begin{bmatrix} X^{REL} \\ Y^{REL} \\ Z^{REL} \\ V_{XT}^I \\ V_{YT}^I \\ V_{ZT}^I \end{bmatrix}$$

The dynamics equations, in vector form, are

$$\frac{d}{dt} \begin{bmatrix} x^{REL} \\ y^{REL} \\ z^{REL} \\ v_{XT}^I \\ v_{YT}^I \\ v_{ZT}^I \end{bmatrix} = \begin{bmatrix} v_{XT}^I - v_{XF}^I \\ v_{YT}^I - v_{YF}^I \\ v_{ZT}^I - v_{ZF}^I \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} \quad (36)$$

Physically, this dynamics model is saying that the target is flying in a basically straight line in the inertial NED frame. The additive noise allows for small target maneuvers and prevents the elements of the error covariance matrix, and hence the gain matrix in the update equations, from going to zero.

The measurement update equations for this algorithm are given by Equation (33), where the target velocities in the relative NED frame are calculated by subtracting fighter inertial velocity from target inertial velocity, so that

$$v_{XT}^{REL} = v_{XT}^I - v_{XF}^I \quad (37a)$$

$$v_{YT}^{REL} = v_{YT}^I - v_{YF}^I \quad (37b)$$

$$v_{ZT}^{REL} = v_{ZT}^I - v_{ZF}^I \quad (37c)$$

The bias correction terms given by Equation (34) and (35) are also used with algorithm three. The other matrices needed for implementation are given in Appendix C; \underline{Q} and \underline{R} are discussed in Section 3.4.

3.3.5 Algorithm Four. The state vector of algorithm four is made up of range R^S in the spherical coordinate system, target total velocity, target heading, and target heading rate. The target motion is considered to be entirely in the plane of the target's local horizon; that is, the target is assumed not to be changing its altitude. The reasoning behind this choice of states, as discussed in the overview, is to provide the filter with a target model with maneuver capability somewhere between the very flexible model of algorithms one and two and the very constrained models of algorithms three, five, and six. Physically, the target's total velocity will not change very quickly; most of the acceleration will be perpendicular to the target's flight path, and mostly in the horizontal plane (the target is performing constant airspeed horizontal turns).

$$\begin{bmatrix} R^S \\ V \\ \alpha \\ \dot{\alpha} \end{bmatrix}$$

where V = target total velocity

α = target heading, measured the same was as θ

$\dot{\alpha}$ = target heading rate

Figure 3-4 shows the geometry of the intercept looking down from above the two aircraft. Because of the measured elevation difference between the target and the fighter (γ , measured as described in Section 3.3.3, Figure 3-3) and the curvature of the earth, there is an angular difference between the local horizontal of the target (the assumed plane of motion) and the local horizontal of the fighter (with respect to which the measurements are taken) which can be approximated by $\gamma + R^S/R_e$, where R_e is the radius of the earth.

From Figure 3-4, it can be seen that the component of the target's velocity which causes a positive change in R^S is $V \cos(\pi - \alpha + \theta) \cos(\gamma + R^S/R_e)$. The dynamics equations can thus be written

$$\frac{d}{dt} \begin{bmatrix} R^S \\ V \\ \alpha \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} V \cos(\pi - (\alpha - \theta) \times 10^{-6}) \cos(\gamma \times 10^{-6} + R^S/R_e) - V_{XF}^{LOS} \\ 0 \\ \dot{\alpha} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ W_1 \\ 0 \\ W_2 \end{bmatrix} \quad (38)$$

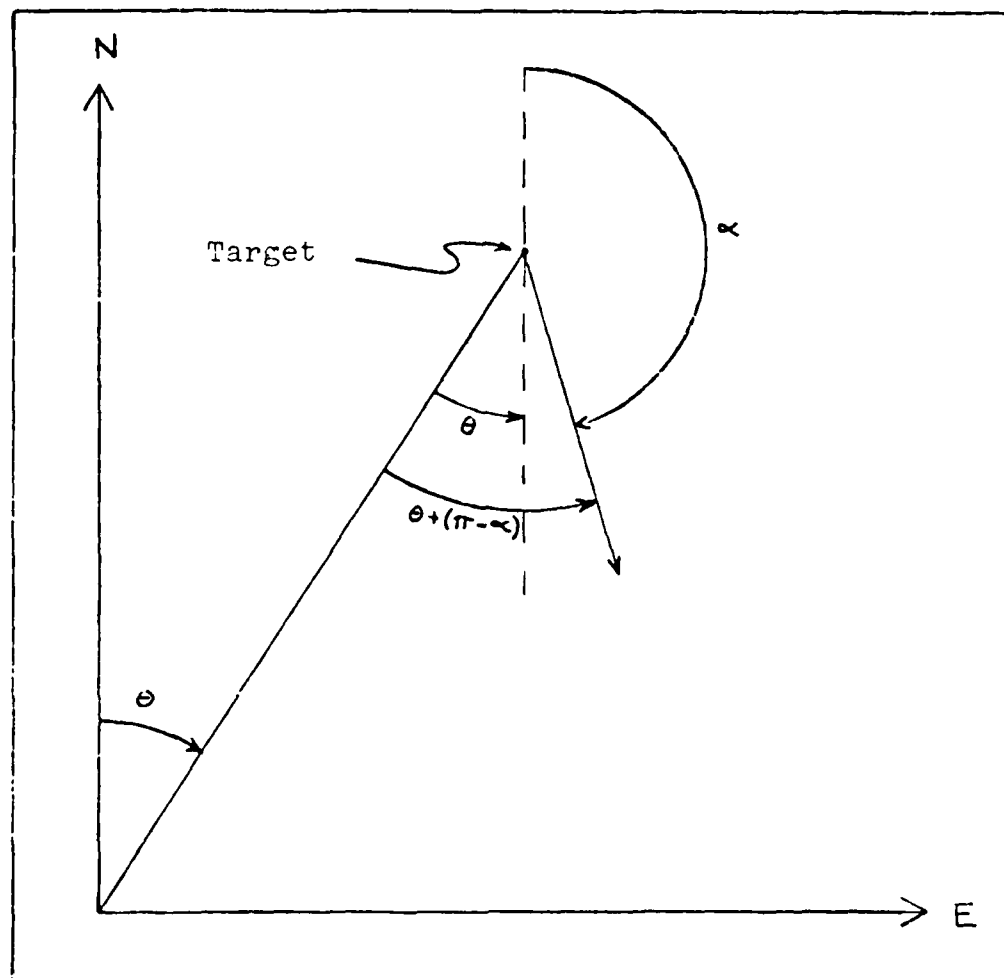


Figure 3-5. Algorithm Four Horizontal Intercept Geometry

where, as previously defined, V_{XF}^{LOS} represents fighter inertial velocity in the X^{LOS} direction, or along the line of sight. The factors of 10^{-6} in Equation (38) are due to expressing the angles in microradians.

The measurements used in this algorithm are azimuth rate and elevation rate; it is again assumed that the azimuth and elevation angle measurements are accurate enough that they can be used directly in the propagation and update equations, which again suggests the possible addition of pseudonoise, although that was not done here. In implementation, in order to minimize errors caused by the changing of θ and γ during the propagation period, the angles used as constants for the entire time of propagation (T) are calculated in the following manner:

$$\theta' = \theta + 0.5T\dot{\theta} \quad (39a)$$

$$\gamma' = \gamma + 0.5T\dot{\gamma} \quad (39b)$$

where θ , γ , $\dot{\theta}$, and $\dot{\gamma}$ are the angle and angle rate measurements at the beginning of the propagation period. Other possibilities exist for correcting this error, but this is a simple method often used to enhance Euler integrations considerably with little expense.

The measurement equations for this algorithm can be written as

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \frac{(V \sin(\pi - (\alpha - \theta) \times 10^{-6}) - V_{YF}^{LOS})}{R^S \cos(\gamma \times 10^{-6} + R^S/R_e)} \times 10^6 \\ (V \cos(\pi - (\alpha - \theta) \times 10^{-6}) \sin(\gamma \times 10^{-6} + R^S/R_e) - V_{ZF}^{LOS}) \times 10^6 + \omega_C \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (40)$$

where ω_C is defined by Equation (27b).

The remaining matrices required for implementation are found in Appendix D; the \underline{Q} and \underline{R} matrices are discussed in Section 3.4.

3.3.6 Algorithm Five. Algorithm five uses a state vector composed of range in the spherical coordinate frame and target velocities in the inertial NED frame. This state vector can be written as

$$\begin{bmatrix} R^S \\ V_{XT}^I \\ V_{YT}^I \\ V_{ZT}^I \end{bmatrix}$$

The component of target velocity which would cause a positive change in R^S can be written as $(V_{XT}^I \cos \theta - V_{YT}^I \sin \theta) \cos \gamma +$

$V_{ZT}^I \sin \gamma'$, where θ' and γ' are θ and γ expressed in radians.

The dynamics equations in vector form are

$$\frac{d}{dt} \begin{bmatrix} R^S \\ V_{XT}^I \\ V_{YT}^I \\ V_{ZT}^I \end{bmatrix} = \begin{bmatrix} (V_{XT}^I \cos \theta' + V_{YT}^I \sin \theta') \cos \gamma' + V_{ZT}^I \sin \gamma' - V_{XF}^{LOS} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ W_1 \\ W_2 \\ W_3 \end{bmatrix} \quad (41)$$

The measurement equations are, as before, noise corrupted version of $\dot{\theta}$ and $\dot{\gamma}$:

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \frac{(V_{YT}^I \cos \theta' - V_{XT}^I \sin \theta' - V_{YF}^{LOS}) \times 10^6}{R^S \cos \gamma'} \\ \frac{(-V_{ZT}^I \cos \gamma' - (V_{XT}^I \cos \theta' + V_{YT}^I \sin \theta') \sin \gamma' - V_{ZF}^{LOS}) \times 10^6}{R^S} + \omega_c \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (42)$$

A variation of this algorithm was used to enhance convergence during the testing of this and the following algorithms. A similar variation would probably help the previous algorithms also, but was not included simply because they were not tested in the detail of algorithms five and six. This variation adds noise ω_4 to the first channel of the filter, in addition to the noise in the velocity channels, thus

making the noise vector in Equation (41)

$$\begin{bmatrix} w_4 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

The reasoning behind this change will be discussed in great detail in Chapter 5, which covers the results of the testing.

Selection of values for Q and R is explained in Section 3.4; the rest of the matrices needed for implementation are given in Appendix E.

3.3.7 Algorithm Six.(3) This method is very similar to algorithm five, the only difference being in the frame of reference in which target velocity is expressed. In algorithm six, target velocity is written in the LOS frame, giving a state vector of

$$\begin{bmatrix} R^S \\ v_{XT}^{LOS} \\ v_{YT}^{LOS} \\ v_{ZT}^{LOS} \end{bmatrix}$$

Because target velocity is already written in the LOS frame, which is aligned with R^S , the dynamics and measurement equations are relatively simple. The dynamics equations are

$$\frac{d}{dt} \begin{bmatrix} R^S \\ V_{XT}^{LOS} \\ V_{YT}^{LOS} \\ V_{ZT}^{LOS} \end{bmatrix} = \begin{bmatrix} V_{XT}^{LOS} - V_{XF}^{LOS} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ W_1 \\ W_2 \\ W_3 \end{bmatrix} \quad (43)$$

The measurement equations are noisy versions of $\dot{\theta}$ and $\dot{\gamma}$:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} [(V_{YT}^{LOS} - V_{YF}^{LOS})/R^S \cos(\gamma \times 10^{-6}) \times 10^{-6}] \\ [(V_{ZT}^{LOS} - V_{ZF}^{LOS})/R^S] \times 10^6 + \omega_C \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (44)$$

As in the previous algorithm noise was added to the range channel at specified times to aid convergence. See Chapter 5 for a further discussion of this.

This algorithm is simpler than algorithm five, but the problem is that while algorithm five models target motion as a straight line in inertial NED coordinates, algorithm six models target motion as a straight line in LOS coordinates. If the target is truly flying a constant heading, the algo-

rithm five model is the more accurate. Based on the mathematical models, one would expect the algorithm six model to be approximately as good as algorithm five in scenarios in which the angle rates are close to zero, such as when the range to the target is large or when the fighter is on approximately a collision course with the target. The relative advantages of these two approaches will be discussed further in Chapter 5.

The \underline{G} , \underline{H} , and \underline{F} matrices needed to implement this algorithm are given in Appendix F; the \underline{Q} and \underline{R} matrices are discussed in the Section 3.4.

3.4 Determination of Values for $\underline{Q}(t)$, $\underline{R}(t_i)$, and τ

This section explains the initial selection of values for the $\underline{Q}(t)$ matrix associated with the dynamics noise, the measurement noise covariance matrix $\underline{R}(t_i)$, and τ , the time constant used in algorithms one and two. The noise $\underline{W}(t)$ is input to the dynamics equations of the six algorithms in one of two ways: as the input to a first order lag (algorithms one and two), or as the input to an integration with no feedback (algorithms three through six). The determination of $\underline{Q}(t)$ for these two cases is discussed separately in Section 3.4.1 and 3.4.2. In both cases, the $\underline{Q}(t)$ obtained is only a reasonable first guess at what $\underline{Q}(t)$ should be; this first guess may be further tuned during the testing process, as explained in Chapter 4 and 5.

3.4.1 Determination of $\underline{Q}(t)$ and τ for Algorithms One and Two. In algorithms one and two, where acceleration is

modelled as the output of a first order lag driven by white Gaussian noise, values selected for $\underline{Q}(t)$ and τ depend on the types of maneuvers anticipated from the target. In general, one would expect a fighter-type aircraft target to have a small τ and large diagonal values in $\underline{Q}(t)$, when compared to a bomber or transport target. The fighter aircraft is capable of higher acceleration (large $\underline{Q}(t)$) and of changing its acceleration more quickly (small τ).

Because the $\underline{Q}(t)$ associated with a particular anticipated target acceleration is a function of τ , a value for τ must first be selected. In this thesis, τ will be set equal to 1.0 seconds. This τ approximates the types of reactions expected from a fighter type target. For a scalar, first order lag shaping filter, it can be shown (5:185) that the appropriate noise strength $\underline{Q}(t)$ to provide a stationary shaping filter output of mean squared value σ_a^2 in steady state is

$$\underline{Q}(t) = Q = 2\sigma_a^2/\tau \quad (45)$$

Assume, for example, that the target is modelled such that, with a probability of .998, it will pull no more than 5 g's, where 1 g = 32.16 ft/sec². If the acceleration is assumed Gaussian, .998 corresponds to a 3σ value, and σ_a^2 can be calculated as $[\frac{5}{3}(32.16)]^2 = 2875 \text{ ft}^2/\text{sec}^4$. In this case, equation (45) yields $\underline{Q}(t) = 5750 \text{ ft}^2/\text{sec}^5$.

The dynamics driving noises on the acceleration channels are assumed to be uncorrelated, which physically says that acceleration in one direction does not depend at all on the

acceleration in either of the other directions. This is not truly representative of probable target motion; there actually are correlations which we are assuming to be zero because of the difficulty in computing accurate off-diagonal terms. Therefore, assuming independent driving noises, the $\underline{Q}(t)$ matrix for algorithms one and two is a 3x3 diagonal matrix with the computed values for the scalar $\underline{Q}(t)$ on the diagonal.

3.4.2 Determination of $\underline{Q}(t)$ for Algorithms Three Through Six. Algorithms three through six model target motion such that acceleration is itself zero, plus a small amount of pseudonoise which is zero-mean, Gaussian, and white, or

$$\dot{V}(t) = 0 + W(t) \quad (46)$$

where $V(t)$ represents the scalar velocity in one direction. It can be shown that, for this type of model, given $E[V^2(t_0)] = 0$, (5:184)

$$E[V^2(t)] = Q \cdot (t-t_0) \quad (47)$$

where $Q = Q(t) = \text{constant strength of the noise } W(t)$.

Thus, to find a value for Q , we simply divide the desired variance in velocity by the time over which the variance occurs. Physically, we are saying that the change in the target's velocity over a one second interval is modelled as a Gaussian random variable with mean zero and variance of $Q \cdot (t-t_0) \text{ ft}^2/\text{sec}^2$.

For example, assume that the target velocity in a certain direction will change over a one second interval by less than 100 ft/sec with a probability of .998. As seen previously,

we can find a value for σ_v of 33.33 ft/sec, so that $Q = 1111.11 \text{ ft}^2/\text{sec}^3$.

An alternate method would be to choose a value for the Q of a stationary white noise so that this height matches the height of the observed power spectral density of the time correlated acceleration at low frequency. This method involves a knowledge of power spectral densities and will not be fully explained here.

It should be noted again that this is a first guess at a value for $Q(t)$, and it may be revised during subsequent tuning.

3.4.3 Determination of $R(t_1)$. The $R(t_1)$ matrix is perhaps the easiest to find and to understand physically. In all algorithms it is assumed that the measurement noises are independent, white, Gaussian, zero-mean, discrete-time noise processes. The assumed independence of the noises physically means that an error in any measurement will have no correlation with an error in any other measurement. It seems reasonable that this would be the case in separating azimuth errors from elevation errors; and, if rates are measured in another way than differentiating the angle measurements (a rate gyro, for example), it also seems reasonable that angle measurements would be independent from their corresponding rate measurements.

As in the case of the dynamics noise, this is knowingly mismodelling the measurement model, as there undoubtedly are off-diagonal terms in the actual $R(t_1)$ matrix. However, due to the difficulty of finding these off-diagonal terms, it is

assumed that $\underline{R}(t_1)$ will always be diagonal. Further, the values of the diagonal components of $\underline{R}(t_1)$ can be set as the variance of the associated elements of the measurement noise vector. For example, if an element of the $\underline{V}(t_1)$ vector is a noise with standard deviation $\sigma = 1000 \text{ } \mu\text{rad/sec}$ being added to a bearing rate measurement, then the associated diagonal term of $\underline{R}(t_1)$ would be $10^6 \text{ } \mu\text{rad}^2/\text{sec}^2$.

In the evaluation of the six algorithms, $\underline{R}(t_1)$ will be varied to analyze the effect of sensor accuracy on algorithm performance. The diagonal terms of $\underline{R}(t_1)$, however, will always reflect the σ^2 value of the measurement errors in the sensor, which are assumed to be given. Thus there is never any mismodelling between the filter $\underline{R}(t_1)$ and the true $\underline{R}(t_1)$. This mismodelling, and the robustness of the filter in handling such mismodelling, is not addressed in this thesis.

CHAPTER IV

METHOD OF EVALUATION

4.1 The Monte Carlo Simulation

In evaluating the performance of a particular algorithm, what we are really looking for is the statistical behavior of the errors in the EKF--generated state estimates as a function of time. If the dynamics and measurement equations were all linear, this could be done via a covariance analysis in which specific measurement realizations need not be simulated; however, since all algorithms evaluated in this study have nonlinearities in either the dynamics or the measurements, we must resort to the ensemble averaging of many test runs to obtain the statistics of the error process. This procedure is known as a Monte Carlo simulation (5:335-341).

Figure 4-1 shows the overall structure of the Monte Carlo analysis. The truth model generates an accurate representation of the signal process \underline{z} , or, in other words, the true measurements. In this thesis, the truth model is based on data generated by a computer program called PROFGEN(8). PROFGEN creates two files of trajectory data, one for the fighter and one for the target, each containing aircraft position, velocity, and acceleration information at 1/15 second intervals. The trajectories are deterministic and based on parameters input to PROFGEN (see Chapter 5 for specific trajectories evaluated). The trajectory data is read by the truth model, which computes the target position and motion relative to the fighter, com-

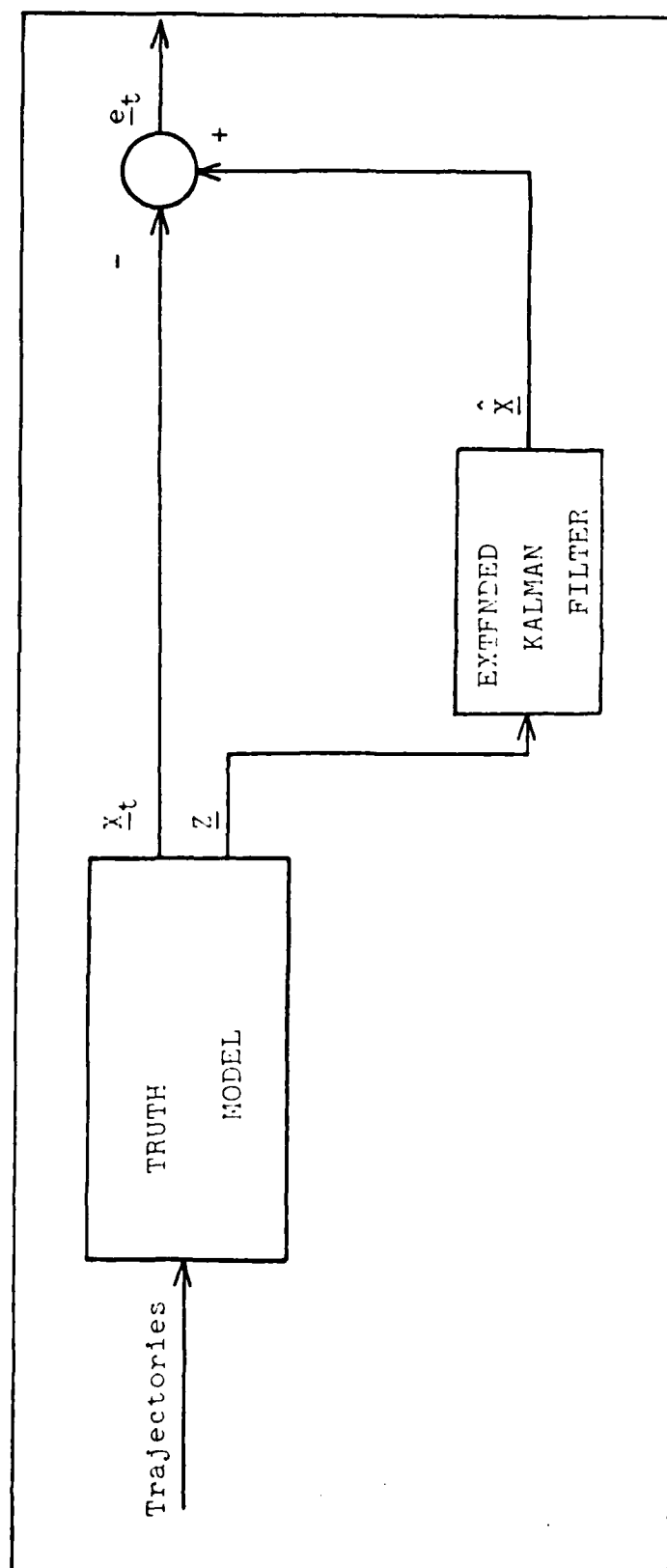


Figure 4-1. Performance Evaluation

puts the true measurements, corrupts these measurements with noise, and hands the noise-corrupted azimuth and elevation (for algorithms two and three), and azimuth rate and elevation rate (for all algorithms) to the filter as measurements. Thus the \underline{z} being input to the EKF in Figure 4-1 is a simulation of an actual measurement at a given time for the given two trajectories. The Kalman filter algorithm processes these simulated measurements and generates an estimate of the target state vector, $\hat{\underline{x}}$, which, when differenced with the true target state vector, \underline{x}_t , yields one realization of the error process at a particular time, $\underline{e}_t(t_1)$. (In the cases evaluated here, the truth model generates a truth vector \underline{x}_t which is the same dimension as \underline{x} , thus permitting direct differencing; this is not generally the case (5:326-336). Additionally, it should be noted that the Kalman filter software implementation used the fifth-order Kutta-Merson integration routine found in SOFE(9) to integrate the state vector and covariance matrix propagation equation, even in the case of linear dynamics.) The Monte Carlo analysis takes the ensemble average of a number of runs to determine the sample statistics of the error process, \underline{e}_t .

In this thesis, the statistics calculations and plotting are done by a software program entitled SOFEPL(10). The plots generated by SOFEPL contain the mean of the error, the mean of the error plus and minus one standard deviation of that mean, and the positive and negative square root of the appropriate diagonal term of the filter-generated error co-

variance matrix (see Figure 4-2). This type of plot gives a good indication of the filter's performance and makes tuning the filter relatively straightforward. In general, the mean error curve should remain within the $\pm \sqrt{P_{11}}$ envelope, and the mean error plus and minus one standard deviation curves of the errors should roughly match the $\pm \sqrt{P_{11}}$ curves (refer to Figure 4-2).

Ten runs are made in each Monte Carlo analysis. This number was chosen by comparing the results of the analysis when using 5, 10, and 20 runs, looking for the point where the computed statistics remain the same as the number of runs is increased. In an effort to conserve computer resources, and because there was very little difference between the results from 10 runs and the results from 20 runs, it was decided that 10 runs would be sufficient.

4.2 Sampling Rates and Sensor Noises Tested

In order to limit the scope of this study to within reasonable bounds, the ranges of the measurement sample rate and the sensor noises to be tested had to be limited. The sample rate was held constant at 1 Hz throughout the tests. This rate is reasonable if the sensor is in a full track mode. The sensor noise standard deviations vary from 1 milliradian to 10 microradians in angle and from 200 microradians per second to 2 microradians per second in angle rate, also providing a physically reasonable set of characteristics.

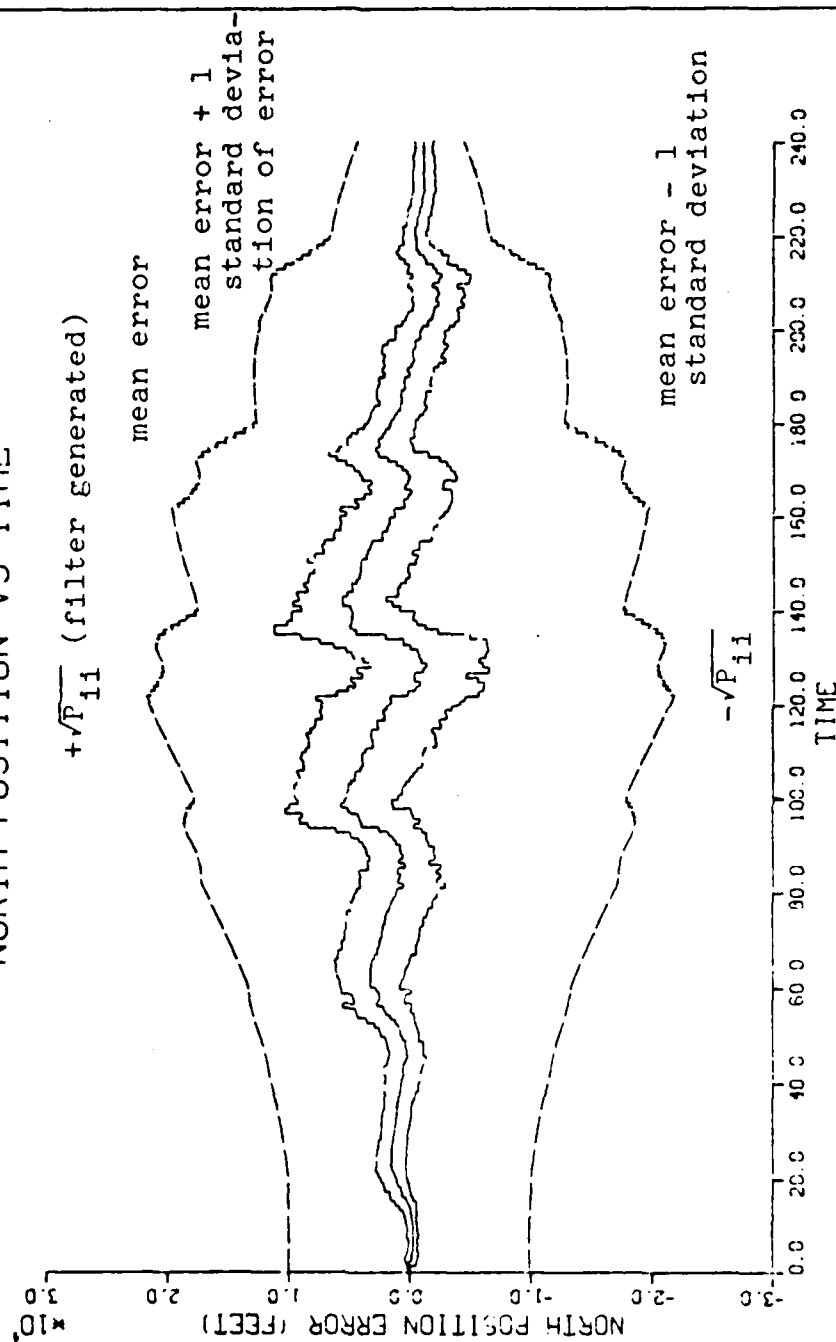


Figure 4-2. Example of SOFEPL Output

4.3 Figures of Merit

When comparing the performance of several algorithms, the criteria used to determine which is "best" must be determined. For this study, the problem was broken down into two phases: convergence and tracking. A dividing time of 40 seconds into the run was used to separate the two phases, thus giving the algorithm a fair amount of time to converge before tracking analysis is begun.

The following figures of merit are proposed:

- 1) the amount of time needed for the mean error to converge from erroneous initial conditions to within 10% of the true range,
- 2) the maximum mean error in range and the maximum magnitude of (mean error plus or minus one standard deviation) after the 40 second convergence time,
- 3) the time average of the mean error and the time average of the standard deviation of the mean error, again after the 40 second convergence time.

The time to converge should give a meaningful measure of acquisition ability, something that is of crucial importance in a ranging algorithm. By including data on the maximum mean error and standard deviation as well as the time average of the mean error and standard deviation, we can portray to some degree how much the errors varied from one simulation run to the next (time averaged standard deviation) as well as how much the errors varied during the course of the runs (maximum values). Said another way, one set of criteria pre-

sents RMS error by averaging out the time varying variability, and a second set of criteria indicates worst deviations from this kind of performance.

Because the Monte Carlo simulation results are valid only for the particular target and fighter trajectories used during any one set of runs, the algorithm comparisons must likewise be qualified as being true only for one particular engagement scenario.

CHAPTER V

RESULTS

5.1 Introduction

The results of the Monte Carlo simulation of the six ranging algorithms are discussed in this chapter. This first section explains the specific fighter and target maneuver profiles which were evaluated and provides some insight into why these profiles were chosen. The evaluation results are examined in the subsequent four sections, in which the algorithms are presented generally in the order of least successful to most successful.

5.2 Maneuver Profiles

5.2.1 Fighter Profile. The maneuver profile flow by the fighter air craft in all simulations is shown in Figure 5-1. It is made up of a sequence of flight segments, each consisting of a fifty degree heading change, a fifty degree heading change in the opposite direction (back to the original heading), and a segment of level, unaccelerated flight. The turns last for ten seconds each, while the straight segment lasts for thirty seconds. The fighter altitude is 25,000 feet, its total velocity is a constant 1000 feet/second (approximately Mach 1), its initial heading is due north, and all turns are accomplished at an acceleration of 3 g's, where $1\text{ g} \approx 32.2\text{ feet/second/second}$.

The altitude and velocity of the fighter were chosen as representative of the altitude and velocity at which an in-

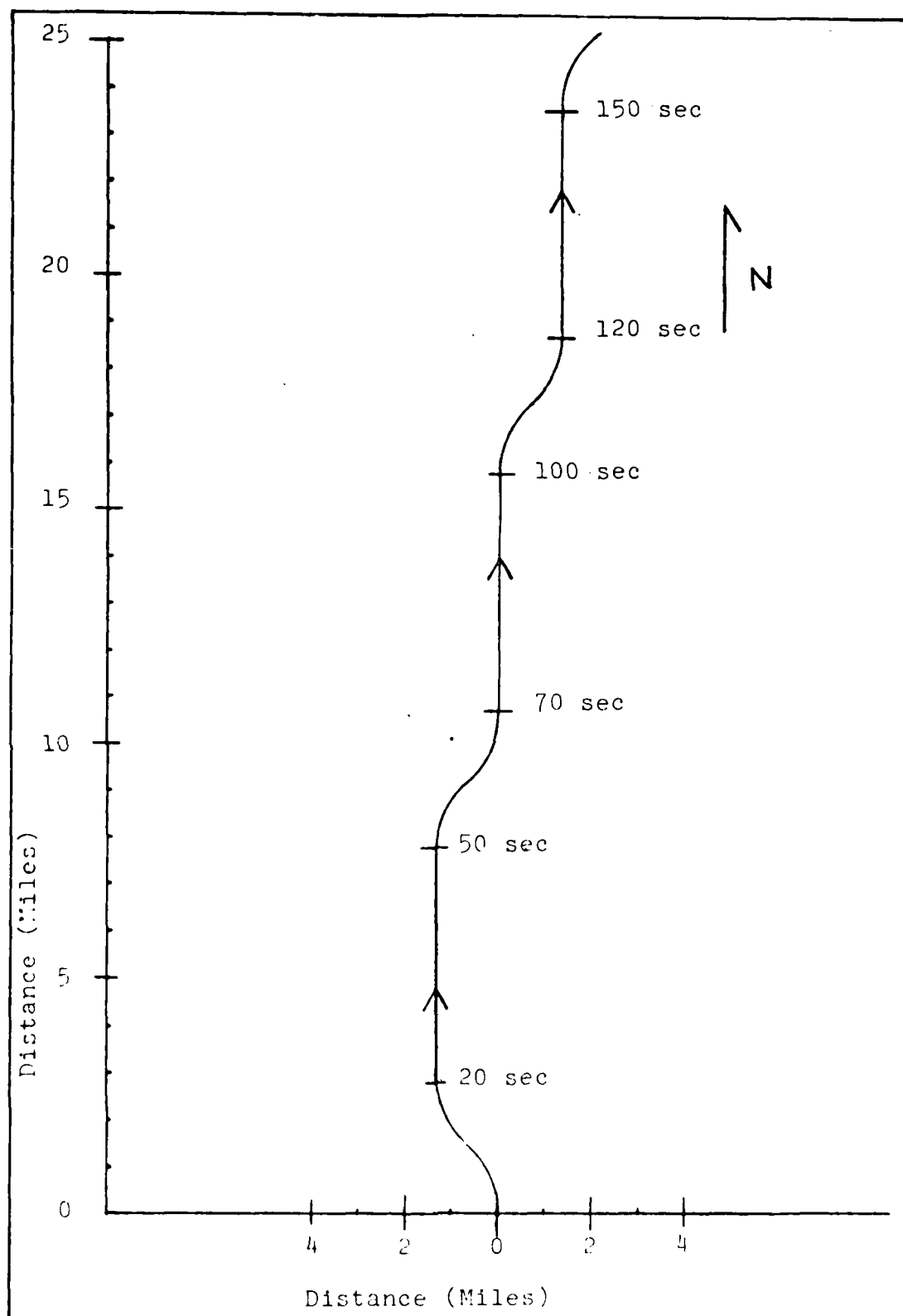


Figure 5-1. Fighter Maneuver Profile

Table 5-1 Target Maneuver Profiles

PROFILE	INITIAL VELOCITY (FT/SEC)	INITIAL HEADING (DEGREES)	INITIAL ALTITUDE (FT x 10 ³)	TIME OF MANEUVER INITIATION (SEC)	TYPE OF MANEUVER
1	1000	180	25	N/A	N/A
2	1000	090	25	N/A	N/A
3	1000	360	25	N/A	N/A
4	2000	180	50	N/A	N/A
4A	2000	180	50	100	5g, 90° left turn
4B	2000	180	50	120	5g, 90° left turn
4C	2000	180	50	100	2g push over to 4° nose low
5	2000	090	50	N/A	N/A
5A	2000	090	50	100	5g, 90° right turn
5B	2000	090	50	120	5g, 90° right turn
6	2000	360	50	N/A	N/A
7	3000	180	50	N/A	N/A
8	3000	090	50	N/A	N/A

terceptor aircraft is likely to be flying in an air defense type scenario. The g loading used in the turns was selected as a moderate value which gives a reasonably rapid turn rate to enhance observability (as discussed in the following paragraph), while still providing for pilot comfort.

The weave type maneuver schedule flown by the fighter was necessitated by an observability problem which is inherent in bearing-only ranging without purposeful direction changing by the attacker (4;11). While lack of observability of target range has not been mathematically shown for three dimensional aircraft motion in the same detail as for the case in which all motion is limited to a single plane, the observability enhancement provided by fighter aircraft maneuver can be seen readily in the simulation results. This point will be discussed further in subsequent sections.

5.2.2 Target Profiles. Thirteen target profiles were used in the evaluation of the ranging algorithms. Profiles 1 through 8 are simulations of non-maneuvering targets, in which the initial target heading, range, velocity, and altitude are varied between profiles. Profiles 4A, 4B, 4C, 5A, and 5B are variants of profiles 4 and 5 in which the target maneuvers at approximately 100 seconds into the run. A summary of the target maneuver schedules is given in Table 5-1. In all cases, the target's initial position is due north of the fighter (on the fighter's nose).

From Table 5-1 it can be seen that the general trend is for target airspeed and altitude to increase as the initial

range increases. The reasoning behind this is that, due to sensor limitations, a target acquired at long range will more likely be a fast, high-flying aircraft, while one acquired at a shorter range is more likely to be at medium altitude and airspeed.

The maneuvers flown in profiles 4A, 4B, 5A, and 5B are 5 g, 90 degree turns. The turns are initiated at 100 and 120 seconds into the run in order to evaluate the effect of target maneuvering both during and after the fighter's observability maneuver, which occurs between 100 and 120 seconds (see Section 5.2.1). The maneuver in profile 4C is a 4 degree push over to evaluate the algorithm's sensitivity to an altitude change. The 4 degree descent will allow the target to descend from 50,000 feet to approximately 25,000 feet during the last 200 seconds of the run.

5.3 Evaluation of Algorithms One and Two

Algorithms one and two (see Section 3.3.2 and 3.3.3) were evaluated against target profile 7, where the algorithms were given perfect initial conditions, and a $\underline{Q}(t)$ matrix with diagonal elements indicative of a target model with a 3g acceleration of 5 g's. The 1σ values used for the angle and angle rate measurement errors were 100 microradians and 20 microradians per second, respectively. The diagonal elements of the $\underline{P}(t_0)$ matrices are given by Table P-1, Appendix P.

The above evaluate conditions were chosen in order to evaluate the algorithms under a scenario favorable to algorithm performance. Target profile 7 is a non-maneuvering profile

and is initiated at a long enough range to give the algorithm time to recover from any initial transients (see Table 5-1). The $Q(t)$ matrix is representative of a relatively benign target, which is accelerating at 1.66 g's or less 68% of the time. By giving the algorithm perfect initial conditions, any problems in the initial convergence caused by an erroneous $\hat{x}(t_0)$ are avoided. The 1 σ error values for angle measurements (used in algorithm two) and angle rate measurements (used in both algorithms) were chosen as typical of what one might expect of a future sensor system. Finally, the error covariance matrix was initialized as a diagonal matrix, with the diagonal terms set to what was considered reasonable σ^2 values for the states they represent, in the case where $\hat{x}(t_0)$ was not known a priori.

As can be seen from the plots in Appendices G and H, both algorithms showed significant range errors, even in the favorable cases evaluated. Although Figure G-1 indicates that further filter tuning may have improved algorithm one's performance to some extent, the basis for the poor performance of both algorithms lies in the observability problem discussed in Section 5.2.1 as it relates to the target model. The Kalman filters were both able to find a solution which met all of the given constraints, so that the residuals were in fact approximately zero mean with the appropriate variances. Due to the lack of observability of all target states for the given target model, however, the algorithm estimates were both severely biased, with mean errors in the position or range states considerably larger than the filter-computed

standard deviation. In other words, there were not enough constraints in the model for target motion given to the filter to allow the filter to converge to the correct answer.

Referring first to Appendix H, it can be seen that algorithm two diverged within the first twenty seconds of the run. By looking at the output of a single run, it was found that the velocity in the north direction was estimated to be a very large negative value. Before this estimate could be corrected, the estimated target had "flown past" the fighter, producing azimuth residuals on the order of $\pi \times 10^6$ micro-radians, i.e., π radians, and resulting in divergence.

Appendix G shows that algorithm one was making an effort to converge to the correct solution, but was making range errors averaging approximately 45 nautical miles throughout the run. One can see clearly from Figures G-1 and G-2 the effects that the observability enhancement maneuvers performed by the fighter every 50 seconds have on aiding filter convergence and decreasing the filter's estimate of the standard deviation of the errors in its state vector estimate.

It is unclear exactly why algorithm two diverged so rapidly while algorithm one appears to be at least making an effort to track the target. However, because neither algorithm appeared to be performing satisfactorily, they were not investigated further.

5.4 Evaluation of Algorithm Three

Algorithm three (see Section 3.3.4) was evaluated against target profile 7, where perfect initial conditions were again

given to the filter. The 1σ values of the angle and angle rate measurement errors were set at 100 microradians and 20 microradians per second. The diagonal terms of $\underline{Q}(t)$ were set to 1000 feet²/second³ in the V_x and V_y channels, and 100 feet²/second³ in the V_z channel. As discussed in Section 3.4.4, this models the target as basically non-maneuvering, with less velocity change in the z direction than in the x and y directions. (This is reasonable from the viewpoint of the expected maneuvers of an actual target, which should be mostly in the horizontal plane, and also helps to further constrain the target model). The diagonal element of $\underline{P}(t_0)$ are given in Table P-1, Appendix P. The justification of these choices is to provide an initial evaluation where conditions are favorable for good algorithm performance, as previously discussed in Section 5.3.

This algorithm tracked well compared to algorithms one and two, as shown by the plots in Appendix I. This result was to be expected, since the target motion was constrained in algorithm three to basically unaccelerated flight. By further constraining the target model, we have limited the set of solutions which fit both the dynamics and measurement equation, thus alleviating to some extent the problem of observability. Again, the effects of fighter maneuver on filter convergence can be seen easily, particularly in Figures I-1 and I-4.

These same figures also show clearly another problem which appeared in algorithms five and six as well. Looking at Figure I-1, we see that a transient occurs in the first

three updates causing a mean error of approximately -400,000 feet at 3 seconds into the run. Because the error is plotted as true state minus filter state, this -400,000 feet means that the filter is over-estimating the x position by this amount. Before the transient can die out, however, the filter-computed error covariance collapses to a very small value, thus indicating that the filter "thinks" that its estimate is relatively accurate. Because of this, the filter puts little weight on the measurements, and the recovery is slow.

Because the filter thinks its estimate of position in the x direction is good, it assumes that the change in position estimate, which is really due to the convergence of the estimate to the true value, is caused by target motion. This induces a large error in the velocity estimate in the x direction (Figure I-4), because the filter estimates that the target is coming toward the fighter much faster than it really is.

A solution to this problem would be to force the filter-computed covariances to remain large artificially until the transients have died out. Because of the computational burden caused by computing bias correction terms, the fact that this algorithm requires azimuth and elevation measurements as well as rate measurements, and the promising nature of algorithms five and six, the investigation of algorithm three was stopped here and further tuning was not attempted.

5.5 Evaluation of Algorithm Four

As previously mentioned in Section 3.3.5, the target model used in algorithm four is an attempt to constrain target motion beyond the time-correlated acceleration models of algorithms one and two but not to the point of completely unaccelerated flight as in algorithms three, five, and six. Algorithm four was evaluated against a target flying profile 7, and was given perfect initial conditions, with $\underline{P}(t_0)$ as given in Table P-1, Appendix P. The standard deviations of the measurement errors were set as in the previous two sections. Justification of these conditions is given in Section 5.3.

Figures J-1 through J-4 show the results of algorithm four when using a $\underline{Q}(t)$ whose diagonal terms are 1000 feet²/second³ (for the v channel) and $2 \times 10^7 \mu$ radians²/second³ (for the $\dot{\alpha}$ channel). Physically, these input data tell the filter that the target is not maneuvering very much (the strength of $\underline{w}(t)$ in the $\dot{\alpha}$ channel corresponding to a 1σ $\dot{\alpha}$ change of 0.3 degrees/second²). As can be seen from these plots, the algorithm did well in tracking the non-maneuvering target under these conditions.

The strength of the $\dot{\alpha}$ channel driving noise was then increased to a value corresponding to a 1σ heading rate change of 3 degrees/second², and the simulation was run again on profile 7, this time for only 100 seconds so that the time axis could be expanded horizontally and the plot read more easily. Figures J-5 through J-8 show that, even when using this target model which is relatively benign compared to what

might be expected from an actual target, the filter's solution, while fitting the dynamics and measurement equation, was not the true solution to the problem. Note the dramatic increase in error standard deviation in Figure J-7 at 50 seconds, caused by the fighter maneuver at that time. Again, because this algorithm could not track a non-maneuvering target when using a target model which assumed even a small amount of maneuverability, and because algorithms five and six looked promising, this algorithm was not investigated further.

5.6 Evaluation of Algorithms Five and Six

5.6.1 Introduction. Algorithms five and six (see Section 3.3.6 and 3.3.7) were evaluated fully against all target profiles, both maneuvering and non-maneuvering. This section is divided into subsections which take the reader chronologically through the algorithm development. Filter convergence enhancement, algorithm performance against maneuvering and non-maneuvering targets, and the effects of varying the strength of the sensor measurement errors are discussed in the next four sections.

The values used for the strength of the dynamics driving noise (the diagonal elements of $\underline{Q}(t)$) in the V_x , V_y , and V_z channels were 1000 feet²/second³, 1000 feet²/second³, and 100 feet²/second³, respectively, throughout the evaluation. The smaller strength in the V_z channel, as before, was set both in order to represent expected real world target motion best and because it was thought that any additional constraint on target motion would aid the algorithm performance, given

that the simulated target met that constraint. The strength of the pseudonoise introduced into the range channel was varied, as discussed in the following section, but was set to 4×10^{11} feet²/second for all runs after initial tuning. This value allowed the estimated σ of the range error from the error covariance matrix to increase to approximately 5×10^5 feet just before the measurement update, thus assuring the most rapid recovery from errors within the range of $\pm 5 \times 10^5$, as discussed in the following section.

The standard deviations of the angle and angle rate measurement errors were set to 100 microradians and 20 microradians per second for all runs except those described in Section 5.6.5, where they were explicitly varied to evaluate algorithm sensitivity. As before, these numbers represent the errors anticipated of a future sensor system. It should be noted that, as explained in Section 3.3.6 and 3.3.7, the angle measurements are not a part of the measurement vector, but are input to the filter as if they were known control inputs. The diagonal elements of $\underline{P}(t_0)$ are given in Table P-1, Appendix P.

5.6.2 Filter Convergence Enhancement. One of the first and most important problems encountered was that of getting the filter to converge from erroneous initial conditions. It was assumed that the sensor used would be capable of acquiring targets out to ranges approaching 200 nautical miles. Since no a priori knowledge of target range is possible in an actual situation, an initial range estimate of 100 nautical miles was chosen in order to facilitate acquisition of maxi-

... .

mum range targets as well as short range "pop up" targets. Thus filter evaluation against target profiles 1 through 8 examines convergence from a -80 nautical mile to +70 nautical mile initial error, and includes (in the case of target profile 4) perfect initial conditions.

The problem with the convergence here is similar to that discussed in Section 5.4 in reference to algorithm three; that is, the filter-generated covariance collapses very quickly, indicating (falsely) a good state estimate. Because the filter believes that it has a very good estimate, it places less weight on the measurements, thus hindering convergence, and adversely affecting velocity estimates as well.

Figures K-1 and K-2 show the range error (0-40 seconds) and the x velocity error (0-200 seconds) of algorithm five when run against target profile 7. In this case, the filter was given correct initial velocity (so that the effect of initial range errors could be seen more clearly), but an initial range which was 80 nautical miles short of the true target range. It can be seen that the filter-generated covariance decreases sharply on the second update, while the mean error converges slowly to near zero at 40 seconds.

The proposed solution was to add pseudonoise to the range channel, thus preventing the filter covariance from converging so rapidly. Figure K-3 shows that same profile with pseudonoise of strength 4×10^{11} feet²/second added. The pseudonoise was added for the first 20 seconds in order to take full advantage of the fighter observability enhance-

ment maneuver, which occurs during those 20 seconds. The mean error in this plot is approaching zero after only 7 to 10 seconds. Additionally, by comparing Figure K-2 with Figure K-4, it can be seen that the velocity bias, which was induced by the range errors where no pseudonoise was added, has been greatly reduced by the addition of pseudonoise.

It should be noted that the mean error curve on Figure K-3 is relatively smooth and the standard deviation of the errors relatively small at 10 seconds when compared to the performance both before and after this time. Also note that the mean error plunges to a negative value (indicating an overestimate of range) at 18-20 seconds. The reason behind the apparent recovery and subsequent negative drift from 20-40 seconds is not known; however, the explanation of filter behavior at 18-20 seconds has an impact on the best maneuvers for the fighter to perform and is at the crux of the filter performance evaluation.

To explain, let us look at a two dimensional problem, where bearing rate is simply equal to the relative velocity perpendicular to the line of sight, divided by the range, or

$$\dot{\theta} = \frac{V_T - F_T}{R} \quad (48)$$

where $\dot{\theta}$ = bearing rate

V_T = target velocity perpendicular to the line of sight

F_T = fighter velocity perpendicular to the line of sight

R = range

In the filter application, however, we do not know V_T , but only the filter's estimate of V_T , which can be written as the true V_T plus some error, e_T . Substituting into Equation (48) and solving for the range estimate \hat{R} gives

$$\hat{R} = \frac{V_T + e_T - F_T}{\dot{\theta}} \quad (49)$$

or

$$\hat{R} = \frac{V_T - F_T}{\dot{\theta}} + \frac{e_T}{\dot{\theta}} \quad (50)$$

Thus the range estimate is equal to the true range plus an error term, $\frac{e_T}{\dot{\theta}}$. In order to minimize the error, we have to minimize $\frac{e_T}{\dot{\theta}}$. This minimization can be done by making e_T as small as possible while making $\dot{\theta}$ as large as possible.

Looking back at the example shown in Figure K-3, we see that the plunge in mean error at 18-20 seconds occurs when the fighter is headed very nearly north and $\dot{\theta}$ becomes nearly zero. The errors produced by an erroneous estimate of target velocity cause the mean error to jump, while errors induced by the measurement noise cause an increase in the standard deviation of the errors. The fighter, therefore, in order to reduce errors, should always maneuver to obtain maximum bearing rate, or, in other words, should make his observability enhancement turns in a direction away from any perceived line of sight rate caused by target motion. Also, the pseudonoise in subsequent runs was stopped at 10 seconds

rather than 20 in order to minimize the effect of these errors (both measurement errors and velocity estimate errors) on the range estimate by allowing an earlier collapse of $\underline{P}(t)$ in the position channels.

Initial fighter turns into and away from the line of sight rate were tested against target profile 2; the turns away from the line of sight rate greatly improved algorithm performance. Because in an actual intercept situation the fighter would not want to turn away from the target any more than necessary, (this tactic would lengthen the time to complete the intercept), the initial fighter maneuver was the only one in which the fighter turned in the direction that would increase azimuth rate. The effect of reducing the time in which pseudonoise is added to 10 seconds can be seen in the plots in Appendices L and M.

So far, this discussion has centered on making $\dot{\theta}$ as large as possible; in order to minimize the error term in Equation (50), however, e_T must also be reduced to its smallest value. The ability of the filter to reduce the size of e_T is a function of target heading with respect to the fighter. If the target is heading directly at the fighter, the filter can determine that the target velocity perpendicular to the line of sight is near zero, and e_T is relatively small. If, on the other hand, the target has a large component of velocity perpendicular to the line of sight, this perpendicular velocity is more a function of the filter's estimate of target range, and is more difficult to estimate accurately. Thus

a target flying directly across the line of sight, while increasing θ , is at the same time and by a proportionally larger amount increasing e_T , and decreasing the accuracy of the filter estimate. This point will be discussed further, and results shown, in the following section.

5.6.3 Algorithm Performance Against Non-Maneuvering Targets. Both algorithms were tested against targets flying the non-maneuvering profiles, 1-8. In all cases, the algorithms were given initial conditions of 100 nautical mile range and 2000 feet/second velocity due south. This range and velocity approximate the mean of the anticipated target range and velocity at acquisition, and are thus the best values for $\hat{x}(t_0)$, given no a priori knowledge of target state. It was assumed that the target would be acquired as it came into range, and would thus have a closing velocity with respect to the fighter; this premise led to the selection of an initial velocity due south, or directly toward the fighter aircraft.

The results are given in Appendices L and M, and in Tables 5-2 and 5-3. The appendices are arranged such that Figure L-1 is the algorithm five plot of the same state and target profile as the algorithm six plot in Figure M-1. Thus the 1TH plots in both appendices can be compared directly. (The reader is cautioned that the scale on the ordinate axis may differ between Appendix L and Appendix M.) The simulations were run for different lengths of time for the different target profiles, ranging from 80 seconds for profiles 1-3 to

Table 5-2 Algorithm 5 Performance, Nominal
Error Variances

TARGET PROFILE	MAX MEAN ERROR (NM)	MAX MEAN ERROR+1 σ (NM)	TIME AVERAGE ERROR (NM)	TIME AVERAGE σ (NM)	CONVERGENCE TIME (SEC)
1	-3.38	-4.64	-1.12	0.47	5
2	-37.99	-38.42	-26.53	0.32	OVER 40
3	4.25	5.51	1.74	0.58	5
4	-8.73	-11.96	-3.94	1.01	5
5	-4.44	-5.50	-1.99	0.89	5
6	7.19	10.35	0.77	2.37	10
7	-21.74	-26.90	-9.49	2.43	15
8	59.45	62.19	16.73	2.36	OVER 40

Table 5-3 Algorithm 6 Performance, Nominal
Error Variances

TARGET PROFILE	MAX MEAN ERROR (NM)	MAX MEAN ERROR+1 σ (NM)	TIME AVERAGE ERROR (NM)	TIME AVERAGE σ (NM)	CONVERGENCE TIME (SEC)
1	0.78	1.71	0.46	0.42	5
2	-38.10	-38.46	-19.37	0.26	OVER 40
3	9.30	11.25	3.63	0.80	5
4	-5.65	-8.88	-0.88	1.19	10
5	-6.36	-7.75	-2.34	1.37	5
6	18.73	27.58	4.32	4.15	10
7	-22.08	-27.46	-6.70	2.63	15
8	61.91	64.50	5.00	2.91	OVER 40

180 seconds for profiles 4-6 to 280 seconds for profiles 7-8. In order to improve the legibility of the plots, the range error versus time plots for target profiles 4-8 were broken into two parts: 0-40 seconds, and 40 seconds to the end of the run. This expanded the figure along the time axis in the 0-40 second plot, and along the ordinate axis on the 40 second to end of run plot, making both easier to read.

Tables 5-2 and 5-3 show the maximum mean error (positive or negative), the maximum magnitude of the mean error plus or minus one standard deviation (i.e., the negative mean error minus one standard deviation or the positive mean errors plus one standard deviation, whichever has the greatest magnitude), and the time averages of the mean error and the standard deviation. These statistics are computed from time = 40 seconds to the end time of the run, allowing 40 seconds for filter convergence, and thus represent tracking statistics. Additionally, the approximate convergence time of the mean error to within 10% of the true range is entered in the tables. Because this convergence time is read from the plots, it is only given to the nearest five seconds.

With one minor exception, the performance of the two algorithms in general was very similar. Both performed well, both in initial convergence and in tracking, against the targets either coming due south or going due north (profiles 1, 3, 4, 6, and 7; see Table 5-1 of Section 5.2.2). As stated before, the filter initial conditions assumed a range of 100 nautical miles and a velocity of 2000 feet/second due

south. This good observed performance is due to the fact that the target velocity perpendicular to the line of sight is near zero, and the filter is able to minimize e_T well, as discussed in Section 5.6.2 (See Figure L-1 through L-3, L-7 through L-13, L-18 through L-25, and the corresponding figures in Appendix M.) For profiles 2, 5, and 8, target velocity perpendicular to the line of sight is not essentially zero, in contrast. Against target profile 5, in which the initial condition happened to be the correct target range, we also see rapid convergence. The filter is able, because of the correct initial range, to find the proper velocity on the first measurement update (see Figures L-16 and M-16). However, when evaluated against profiles 2 and 8, where an erroneous initial range led to an erroneous velocity estimate (Figures L-4 through L-6, L-26 through L-29, and corresponding Appendix M plots), the filters are extremely slow in converging. The effect of fighter maneuver on algorithm convergence can be seen clearly in Figure L-6, and, to a lesser degree in Figures L-4 and L-5. The fighter was maneuvering from 50 to 70 seconds into the run, causing a dramatic recovery in the V_y channel. The errors induced in the V_x channel, as previously discussed, are caused by the increase in filter range estimate as it converges toward the true range. Because the filter-generated error covariance in the range channel is small, the filter "thinks" that its range estimate has been accurate since 10 seconds into the run, and it interprets the range estimate change as target motion.

A general trend encountered in both algorithms is the constant negative drift of range and x velocity estimate during the time when the fighter is not maneuvering (Figures L-15, L-19, L-23 and L-24 provide good examples). The drift seems in most cases to be worse at longer range. The cause of this drift is not known; it is assumed that the drift was caused by linearizing the measurement equation.

An example of the difference between the two algorithms can be seen by comparing Figure L-11 with Figure M-11. Because algorithm six assumes that target velocity is (essentially) constant in the LOS frame, the rotation of the LOS frame with respect to the earth biases the algorithm output. This effect is most apparent toward the end of the run, because the target is getting closer to the fighter and the LOS frame is rotating faster. This bias also affects convergence to a small degree, since, as discussed earlier, the best fighter maneuver with regard to convergence is that which gives the highest angle rates, thus providing the worst case for the algorithm six target motion assumptions (see Figures L-10 and M-10).

When these algorithms were first tested, both algorithms diverged when run against target profile 1 (target headed due south from 30 nautical mile initial range), and algorithm five diverged when run against target profile 6 (target headed due north from 100 nautical mile initial range). By examining the output of a single run of the Monte Carlo simulation, it was noted that the range estimate in the state

vector went negative during the initial transient in filter acquisition. Because the filter equations are all based on a positive range, the algorithm could not recover from a negative range estimate.

In order to correct this problem, an artificial lower bound was placed on the range estimate, so that if the range estimate ever went below zero, it was reset to 100,000 feet. The value of 100,000 feet was chosen as relatively short range (about 16.5 nautical miles) from which the filter could converge to the true range with a less violent transient response. While both algorithms were run again against target profile 6, apparently only algorithm five required the artificial lower bound, as there was no change in the algorithm six plots. In the case of algorithm five, Figure L-18 shows that after the second update, the mean of the error is approximately +500,000 feet, indicating a range estimate of 100,000 feet, with zero standard deviation. This is the result of resetting the range estimate (zero standard deviation implies that it was reset on all Monte Carlo simulation runs), and explains the difference between the performance of the two algorithms against target profile 6.

5.6.4 Algorithm Performance Against Maneuvering Targets.

For the following reasons, only algorithm five was tested against target maneuvers:

1. algorithms five and six performed similarly against non-maneuvering targets
2. the difference between the dynamics and measurement equations of the two algorithms leads one to anticipate similar performance against all target profiles

3. an analysis of algorithm six against target profile 4A produced results similar to those of algorithm five against the same target profile

In addition to target profiles 4A, 4B, 4C, 5A, and 5B described in Section 5.2.2, algorithm five was also evaluated against a target which maintained a collision course with the fighter. Appendix N shows the results of these simulations.

Figure N-1 through N-3 show the results of a target maintaining collision course throughout the initial fighter maneuver. It was expected that having an essentially zero bearing rate throughout the maneuver would cause the algorithm to do poorly in estimating range, and that, due to the increased gain in the first 10 seconds, this anticipated effect would be most apparent during the initial maneuver. The filter, however, tracked the changing target velocities well enough to keep the range at approximately its initial value. Thus the filter, lacking other information, assumes that the initial range is correct and changes its estimate of target velocity to fit the received measurement data (the observability problem again being manifested). The cause of the rapid drift from 20 to 30 seconds seen in all three plots could not be determined.

Figures N-4 through N-7 and N-8 through N-11 show filter performance against target profiles 4A and 4B, in which the target is flying south and makes a hard left turn to east at $t=100$ seconds (profile 4A) or $t=120$ seconds (profile 4B). Because the algorithm is getting much of its information during the fighter maneuver, it was anticipated that a target

maneuver during this time (from $t=100$ to $t=120$) would cause a larger error than one initiated after the fighter maneuver was complete. The time of the target maneuver does make a slight difference, but the algorithm performance in both cases shows the same trend. As seen previously, and discussed fully in Section 5.6.2, the filter solves for a range/velocity combination which fits the measurement data, but when the target is moving basically perpendicular to the line of sight, it is difficult to determine the correct range/velocity combination. Thus the error in velocity (e_T in Section 5.6.2) remains large, and the range estimate remains in error by almost 30 nautical miles, converging only very slowly. This error appears as a bias after $t=100$ seconds in Figures N-5, N-7, N-9, and N-11. Note that the recovery of the x velocity in Figures N-6 and N-10 only indicates that the range error (which is still mostly in the north-south direction) has stabilized, not that it has recovered. If one more observability enhancement maneuver were performed, one would expect the errors in the range and V_y channels to be decreased, and the error in the V_x channel to increase in the negative direction as it again misinterprets the range estimate change as target motion.

An examination of Figures N-12 through N-15 and N-16 through N-19 shows that, as expected (from the immediately preceeding discussion and Section 5.6.2), the filter converges to the correct range much faster when the target turns south (target profiles 5A and 5B), thus reducing its velocity component perpendicular to the line of sight. There is, in this

case, a significant difference between plots where the target maneuvered at 100 seconds (during attacking fighter maneuvers), and those where the target maneuvered at 120 seconds (after the fighter maneuver). In comparing Figures N-14 and N-15 against N-18 and N-19, it can be seen that the error in the velocity estimates is less when the target maneuvered after the fighter maneuver was complete. It is not clear why the range estimates in Figures N-13 and N-17 were in error in opposite directions; however, it is apparent that the filter was less in error and converged more quickly when evaluated against target profile 5B.

Finally, Figures N-20 through N-24 show filter performance against a target that begins a shallow descent at $t=100$ seconds (target profile 4C). While filter convergence does not appear to be a problem, it is somewhat surprising that a 4 degree target pitch change can cause a change of approximately 60,000 feet in the range estimate. The range estimate did, however, recover well from this error (see Figure N-21). This maneuver was not evaluated when initiated at a time other than during the fighter maneuver; however, it is assumed that, as in horizontal target maneuvers, filter performance would be degraded less if the target maneuver occurred while the fighter was flying straight and level.

5.6.5 Effects of Varying Measurement Errors. Lastly, runs were made in which the σ values for the angle and angle rate measurements were varied up and down by an order of magnitude from the 100 microradians and 20 microradians/second

used in previous evaluations. As explained in Sections 3.3.6 and 3.3.7, the angle measurements are not a part of the measurement vector, but are considered as a known control input in order to reduce the computational load. The filter $R(t_1)$ matrix was changed corresponding to the change in the errors being fed into the filter, so that the algorithm always "knew" the true variance of the measurement errors. (While robustness to unmodelled changes in the real world was not investigated, it easily could be in the future.) The results of these runs are given in Figures 0-1 through 0-16 (for target profile 5, with the target initially at 100 nautical miles headed east) and Figures 0-17 through 0-32 (for target profile 7, with the target initially at 180 nautical miles headed south) of Appendix 0, and in Tables 5-4 and 5-5 to follow. The plots in Appendix 0 can be compared to runs made previously against the same target profiles (again, 5 and 7) using the nominal measurement error strengths, the results of which are given in Figures L-14 through L-17 and L-22 through L-25.

The results of varying the σ values of the bearing rate measurements, given in Figure 0-1 through 0-8 and 0-17 through 0-24, are exactly as anticipated. Lowering the standard deviation to 2μ rad/sec improves performance both by decreasing the standard deviation of the errors in the filter estimates and by decreasing the mean error. Increasing the σ error value has the opposite effect. One can also see that the filter generated covariance changes in the same manner as the true error variance, thus indicating that the filter is aware

TABLE 5-4 ALGORITHM FIVE PERFORMANCE, TARGET PROFILE 5
VARYING MEASUREMENT ERROR STRENGTHS

SENSITIVITY VARIATION	ANGLE MEASUREMENT ERROR (1 σ) (μ RAD/SEC)	ANGLE MEASUREMENT ERROR (1 σ) (μ RAD)	MAX MEAN ERROR (NM)	MAX MEAN ERROR $\pm 1\sigma$ (NM)	TIME AVERAGE MEAN ERROR (NM)	TIME AVERAGE σ (NM)	CONV. TIME (SEC)
BASE LINE	20	100	-4.44	-5.50	-1.99	0.89	5
PAGE ERROR STRENGTH	2 200	100 100	2.59 -3.51	2.65 -8.38	1.36 -1.55	0.08 3.84	5 10
ANGLE ERROR STRENGTH	20 20	10 1000	-4.42 -4.92	-5.46 -6.54	-1.95 -2.65	0.89 1.17	5 5

TABLE 5-5 ALGORITHM FIVE PERFORMANCE, TARGET PROFILE 7,
VARYING MEASUREMENT ERROR STRENGTH

SENSITIVITY VARIATION	ANGLE MEASUREMENT ERROR (1 σ) (μ RAD/SEC)	ANGLE MEASUREMENT ERROR (1 σ) (μ RAD)	MAX MEAN ERROR (NM)	MAX MEAN ERROR $\pm 1\sigma$ (NM)	TIME AVERAGE MEAN ERROR (NM)	TIME AVERAGE σ (NM)	CONVERGENCE TIME (SEC)
BASE LINE	20	100	-21.74	-26.90	-9.48	2.43	15
PAGE ERROR STRENGTH	2 200	100 100	11.56 -22.06	13.07 -42.37	-2.53 -10.41	0.44 9.56	15 10
ANGLE ERROR STRENGTH	20 20	10 1000	-21.24 -31.09	-26.24 -37.85	-9.28 -15.40	2.47 3.00	15 15

that its state estimates will be less accurate if given less accurate measurements.

Since the angle measurements are brought into the algorithm without being filtered, the results of varying the errors in angle measurements was somewhat less predictable, although one would still expect that a larger error variance would lead to a larger error in the algorithm's estimate. In comparing Figures O-9 through O-12 with L-14 through L-17, and O-25 through O-28 with L-22 through L-25, it can be seen that the difference is so small as to be virtually undetectable. An increase in the 1σ error standard deviation to 1 milliradian causes a significant increase both in the mean and the variance of the errors in the estimates. Note that the filter generated covariance does not change appreciably in this case.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusion

Of the six algorithms examined, only those using non-accelerating target models were able to track a non-maneuvering target successfully. This is as expected, and due for the most part to the observability problem in finding target range from bearing and bearing rate measurements. If the target model in the filter is that of a maneuvering aircraft, the filter is unable to select the correct solution for the target states out of the infinitely large set of solutions which solve the dynamics and measurement equations.

The two algorithms that were fully evaluated against non-maneuvering targets performed similarly against these target profiles. In cases in which the target was flying either directly toward or directly away from the fighter, both algorithms converged quickly and tracked well. In cases in which the target was moving perpendicular to the line of sight, both algorithms experienced difficulty in converging from erroneous initial conditions.

Algorithm five, the only algorithm evaluated against maneuvering targets and with varying measurement error strengths, performed as expected in all cases. When the target maneuvered to decrease its velocity component perpendicular to the line of sight, or turned into the fighter, the algorithm was quick in determining the new target state. When the target maneuver

increased the target velocity across the line of sight, however, the filter experienced difficulty in recovering to the true target state. (As explained in Section 5.6.2, however, if the attacker maneuvers so as to increase $\dot{\theta}$ in Equation (50) without the corresponding increase in e_T seen when the target causes $\dot{\theta}$ to increase, range estimation improves.) Varying the angle rate measurement error standard deviation caused a corresponding variation in the standard deviation of the errors in the filter's estimates. While decreasing the standard deviation of the angle measurement errors had little effect, increasing this standard deviation caused an increase in both the mean and the variance of the errors in the filter's estimate.

6.2 Recommendations

Algorithms five and six should be further evaluated against other target profiles to determine their effectiveness against targets having intermediate heading crossing angles. These target profiles should reflect both realistic scenarios, and geometries and trajectories that are particularly hard or easy for the filter to handle.

The robustness of algorithms five and six to mismodelling measurement errors should be tested. These tests should include cases in which the assumed error variances in $\underline{R}(t_1)$ are greater than and less than the variances of the error in the measurements actually given to the filter.

The possibility of developing a multiple model filter (6:129-136) using variations of algorithm four should be in-

vestigated. If heading rate is input as a parameter, it may be possible to track a maneuvering target using several simplified filters running in parallel, each assuming a different heading rate.

The Bayesian filtering approach discussed by H. W. Sorenson in Reference 12 should also be investigated. This approach was examined briefly in the course of this thesis, but, due to time constraints, was not pursued fully. Appendix Q gives a brief discussion of these efforts.

BIBLIOGRAPHY

1. Aidala, V. J. "Kalman Filter Behavior In Bearings - Only Tracking Applications," IEEE Transactions on Aerospace and Electronic Systems, AES-15: 29-39 (January 1979).
2. Fagerlund, S. Target Tracking Based on Bearing Only Measurements, Technical Report LIOS-R-1003, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, June, 1980.
3. Howard, C. W. "Formulation and Performance Evaluation of a Maneuver Passive Ranging Algorithm," Interdepartmental Correspondence, Hughes Aircraft Corp., 13 April 1979.
4. Lindgren, A. G. and K. F. Gong "Position and Velocity Estimation Via Bearing Observations," IEEE Transactions on Aerospace and Electronics Systems, AES-14: 564-577 (July 1978).
5. Maybeck, P. S. Stochastic Models, Estimation, and Control (Volume 1). New York: Academic Press, 1979.
6. Maybeck, P. S. Stochastic Models, Estimation, and Control (Volume 2). New York: Academic Press, 1982.
7. Mitchell, R. A. K. High Accuracy Aircraft to Satellite Tracking Using an Extended Kalman Filter. MS thesis. Wright-Patterson AFB OH: School of Engineering, Air Force Institute of Technology, 1974.
8. Musick, S. H. PROFGEN - A Computer Program for Generating Flight Profiles, Technical Report AFAL-TR-76-247, November, 1976. (AD A034993).
9. Musick, S. H. SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation, User's Manual, Technical Report, AFWAL-TR-80-1108, October, 1980.
10. Musick, J. H. SOFEPL: A Plotting Postprocessor for 'SOFE', User's Manual, Technical Report, AFWAL-TR-80-1109, November, 1981.
11. Nardone, S. C. and V. J. Aidala "Observability Criteria for Bearings - Only Target Motion Analysis," IEEE Transactions on Aerospace and Electronic Systems, AES-17: 162-166 (March 1981).

12. Sorenson, H. W. The Bayesian Approach to Nonlinear Filtering and Plant Identification. Unpublished paper, presented at the 6th IFAC Symposium, Arlington VA, 7-11 June 1982. Department of Applied Mechanics and Engineering Sciences, University of California, San Diego, May 1982.

APPENDIX A ADDITIONAL MATRICES FOR ALGORITHM ONE

This appendix gives the additional matrices needed to implement algorithm one, in which the state vector is

$$[R^S \quad \dot{R}^S \quad A_{XT}^{LOS} \quad \dot{\theta} \quad A_{YT}^{LOS} \quad \dot{\gamma} \quad A_{ZT}^{LOS}]^T :$$

$$\underline{G}(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (A-1)$$

$$\underline{H}(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (A-2)$$

The $F[t; \hat{x}(t/t_1)]$ matrix is of dimension 7×7 . It's non-zero elements are as follows:

$$F_{12} = 1 \quad (A-3a)$$

$$F_{21} = (\dot{\gamma}^2 + \dot{\theta}^2 \cos^2 \gamma) \times 10^{-12} \quad (A-3b)$$

$$F_{23} = 1 \quad (A-3c)$$

$$F_{24} = 2R^S \dot{\theta} \cos^2 \gamma \times 10^{-12} \quad (A-3d)$$

$$F_{26} = 2R^S \dot{\gamma} \times 10^{-12} \quad (A-3e)$$

$$F_{33} = -\frac{1}{\tau} \quad (A-3f)$$

$$F_{41} = \frac{(A_{YF}^{LOS} - A_{YT}^{LOS}) \times 10^6}{\cos \gamma R^S} + \frac{2\dot{R}^S \dot{\theta}}{R^S} \quad (A-3g)$$

$$F_{42} = \frac{-2\dot{\theta}}{R^S} \quad (A-3h)$$

$$F_{44} = \frac{2\dot{R}^S}{R^S} + 2 \dot{\gamma} \tan \gamma \times 10^{-6} \quad (A-3i)$$

$$F_{45} = \frac{1}{R^S \cos \gamma} \quad (A-3j)$$

$$F_{46} = 2 \dot{\theta} \tan \gamma \times 10^{-6} \quad (A-3k)$$

$$F_{55} = -\frac{1}{\tau} \quad (A-3l)$$

$$F_{61} = \frac{2\dot{R}^S \dot{\gamma}}{R^S} + \frac{(A_{ZF}^{LOS} - A_{ZT}^{LOS}) \times 10^6}{R^S} \quad (A-3m)$$

$$F_{62} = -2\dot{\gamma}/R^S \quad (A-3n)$$

$$F_{64} = -\dot{\theta} \sin 2\gamma \times 10^{-6} \quad (A-3o)$$

$$F_{66} = -2\dot{R}^S/R^S \quad (A-3p)$$

$$F_{67} = 1/R^S \quad (A-3q)$$

$$F_{77} = -\frac{1}{\tau} \quad (A-3r)$$

APPENDIX B

ADDITIONAL MATRICES FOR ALGORITHM TWO

This appendix contains the additional matrices needed to implement algorithm two, in which the state vector is

$$[x^{REL} \quad y^{REL} \quad z^{REL} \quad v_{XT}^{REL} \quad v_{YT}^{REL} \quad v_{ZT}^{REL} \quad A_{XT}^I \quad A_{YT}^I \quad A_{ZT}^I]^T:$$

$$\underline{G}(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (B-1)$$

$$\underline{F}[t; \hat{x}(t_1^-)] = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau \end{bmatrix} \quad (E-2)$$

Because the $\underline{H}[t_1; \hat{x}(t_1^-)]$ matrix can be more easily written using terms defined in the computation of the bias correction terms, the bias correction terms, $\hat{b}_{m1}(t_1^-)$ and $\hat{b}_{m2}(t_1^-)$, will be described next. The two matrices in Equ-

tion (35) which have not yet been specified will be designated \underline{H}^1 and \underline{H}^2 , where

$$\underline{H}^1 \triangleq \frac{\partial^2 h_1[\hat{x}(t_1^-), u(t_1), t_1]}{\partial \underline{x}^2} \quad (\text{B-3})$$

and

$$\underline{H}^2 \triangleq \frac{\partial^2 h_2[\hat{x}(t_1^-), u(t_1), t_1]}{\partial \underline{x}^2} \quad (\text{B-4})$$

Additionally, in an effort to reduce the complexity of the equations in this appendix, R_1 and R_2 will be defined as follows:

$$R_1 \triangleq \sqrt{X_{REL}^2 + Y_{REL}^2} \quad (\text{B-5})$$

$$R_2 \triangleq \sqrt{X_{REL}^2 + Y_{REL}^2 + Z_{REL}^2} \quad (\text{B-6})$$

The \underline{H}^1 and \underline{H}^2 matrices are dimension 9x9, the non-zero elements of which follow. The elements of the state vector are evaluated at time t_1^- .

$$H_{11}^1 = \frac{2X_{REL} Y_{REL}}{R_1^4} \times 10^6 \quad (\text{B-7a})$$

$$H_{22}^1 = \frac{-2X_{REL} Y_{REL}}{R_1^4} \times 10^6 \quad (\text{B-7b})$$

$$H_{12}^1 = \frac{(Y_{REL}^2 - X_{REL}^2)}{R_1^4} \times 10^6 \quad (\text{B-7c})$$

$$H_{21}^1 = \frac{(Y^{REL^2} - X^{REL^2}) \times 10^6}{R_1^4} \quad (B-7d)$$

and

$$H_{11}^2 = Z^{REL} \left[\frac{-2X^{REL^2}}{R_1 R_2^4} - \frac{X^{REL^2}}{R_2^2 R_1^3} + \frac{1}{R_1 R_2^2} \right] \times 10^6 \quad (B-8a)$$

$$H_{12}^2 = X^{REL} Y^{REL} Z^{REL} \left[\frac{-2}{R_1 R_2^4} - \frac{1}{R_1^3 R_2^2} \right] \times 10^6 \quad (B-8b)$$

$$H_{21}^2 = H_{12}^2 \quad (B-8c)$$

$$H_{13}^2 = \frac{X^{REL}}{R_1} \left[\frac{1}{R_2^2} - \frac{2Z^{REL^2}}{R_2^4} \right] \times 10^6 \quad (B-8d)$$

$$H_{22}^2 = Z^{REL} \left[\frac{-2Y^{REL^2}}{R_1 R_2^4} - \frac{Y^{REL^2}}{R_1^3 R_2^2} + \frac{1}{R_1 R_2^2} \right] \times 10^6 \quad (B-8e)$$

$$H_{23}^2 = \frac{Y^{REL}}{R_1} \left[\frac{1}{R_2^2} - \frac{2Z^{REL^2}}{R_2^4} \right] \times 10^6 \quad (B-8f)$$

$$H_{31}^2 = X^{REL} \left[\frac{2R_1}{R_2^4} - \frac{1}{R_1 R_2^2} \right] \times 10^6 \quad (B-8g)$$

$$H_{33}^2 = \frac{2Z^{REL} R_1}{R_2^4} \times 10^6 \quad (B-8i)$$

$\underline{H}[t_1; \hat{\underline{x}}(t_1^-)]$ is a 4x9 matrix, the non-zero terms of which are given below. Again in order to keep the complexity and length of these equations to a minimum, the elements of $\underline{H}[t_1; \hat{\underline{x}}(t_1^-)]$ will be expressed in terms of the elements of \underline{H}^1 and \underline{H}^2 where possible. The elements of the state vector estimate are evaluated at time t_1^- .

$$H_{11} = \frac{-Y^{REL}}{R_1^2} \times 10^6 \quad (B-9a)$$

$$H_{12} = \frac{X^{REL}}{R_1^2} \times 10^6 \quad (B-9b)$$

$$H_{21} = \frac{Z^{REL} X^{REL}}{R_1 R_2^2} \times 10^6 \quad (B-9c)$$

$$H_{22} = \frac{Z^{REL} Y^{REL}}{R_1 R_2^2} \times 10^6 \quad (B-9d)$$

$$H_{23} = \frac{-R_1}{R_2^2} \quad (B-9e)$$

$$H_{31} = V_{XT}^{REL} H_{11}^1 + V_{YT}^{REL} H_{12}^1 \quad (B-9f)$$

$$H_{32} = V_{XT}^{REL} H_{12}^1 + V_{YT}^{REL} H_{22}^1 \quad (B-9g)$$

$$H_{34} = \frac{-Y^{REL}}{R_1^2} \times 10^6 \quad (B-9h)$$

$$H_{35} = \frac{X^{REL}}{R_1^2} \times 10^6 \quad (B-9i)$$

$$H_{41} = V_{XT}^{REL} H_{11}^2 + V_{YT}^{REL} H_{21}^2 + V_{ZT}^{REL} H_{31}^2 \quad (B-9j)$$

$$H_{42} = V_{XT}^{REL} H_{12}^2 + V_{YT}^{REL} H_{22}^2 + V_{ZT}^{REL} H_{32}^2 \quad (B-9k)$$

$$H_{43} = V_{XT}^{REL} H_{13}^2 + V_{YT}^{REL} H_{23}^2 + V_{ZT}^{REL} H_{33}^2 \quad (B-9l)$$

$$H_{44} = \frac{X^{REL} Z^{REL}}{R_1 R_2^2} \times 10^6 \quad (B-9m)$$

$$H_{45} = \frac{Y^{REL} Z^{REL}}{R_1 R_2^2} \times 10^6 \quad (B-9n)$$

$$H_{46} = \frac{-R_1}{R_2^2} \times 10^6 \quad (B-9o)$$

APPENDIX C

ADDITIONAL MATRICES FOR ALGORITHM THREE.

This appendix contains additional matrices needed to implement algorithm three, in which the state vector is

$$[X^{REL} \quad Y^{REL} \quad Z^{REL} \quad V_{XT}^I \quad V_{YT}^I \quad V_{ZT}^I]^T:$$

$$\underline{G}(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (C-1)$$

$$\underline{F}[t; \hat{\underline{x}}(t/t_1)] = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (C-2)$$

The \underline{H}^1 and \underline{H}^2 matrices defined for algorithm three in Appendix B are the same for algorithm three, except that they are now 6x6 rather than 9x9. Similarly, the $\underline{H}[t_1; \hat{\underline{x}}(t_1^-)]$ matrix for algorithm three is a 4x6 matrix with the same non-zero elements as the $\underline{H}[t_1; \hat{\underline{x}}(t_1^-)]$ matrix for algorithm two described in the previous appendix, where the target velocities in the relative NED frame are defined by Equations (37a) - (37c), Section 3.3.3.

APPENDIX D

ADDITIONAL MATRICES FOR ALGORITHM FOUR

This appendix contains additional matrices needed to implement algorithm four, in which the state vector is

$$[R^S \quad V \quad \alpha \quad \dot{\alpha}]^T:$$

$$\underline{G}(t) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (D-1)$$

The non-zero terms of the 4×4 $\underline{F}[t; \hat{x}(t/t_1)]$ matrix are

$$F_{11} = V \cos(\pi + (\theta - \alpha) \times 10^{-6}) \sin(\gamma \times 10^{-6} + R^S/R_e) x \frac{1}{R_e} \quad (D-2a)$$

$$F_{12} = -\cos(\pi + (\theta - \alpha) \times 10^{-6}) \cos(\gamma \times 10^{-6} + R^S/R_e) \quad (D-2b)$$

$$F_{13} = -V \cos(\gamma \times 10^{-6} + R^S/R_e) \sin(\pi + (\theta - \alpha) \times 10^{-6}) \times 10^{-6} \quad (D-2c)$$

$$F_{34} = 1 \quad (D-2d)$$

where R_e = radius of the earth $\approx 20,891,000$ ft.

The non-zero elements of the 2×4 $\underline{H}[t_1; \hat{x}(t_1^-)]$ matrix are

$$H_{11} = (-V \sin(\pi + (\theta - \alpha) \times 10^{-6}) - V_{VF}^{LOS}) \frac{1}{R^S \cos(\gamma \times 10^{-6} + R^S/R_e)} - \frac{\sin(\gamma \times 10^{-6} + R^S/R_e)}{R^S R_e \cos^2(\gamma \times 10^{-6} + R^S/R_e)} \times 10^6 \quad (D-3a)$$

$$H_{12} = \frac{\sin(\pi + (\theta - \alpha) \times 10^{-6})}{R^S \cos(\gamma \times 10^{-6} + R^S/R_e)} \times 10^6 \quad (D-3b)$$

$$H_{13} = \frac{-V \cos(\pi + (\theta - \alpha) \times 10^{-6})}{R^S \cos(\gamma \times 10^{-6} + R^S/R_e)} \quad (D-3c)$$

$$H_{21} = \frac{-V \cos(\pi + (\theta - \alpha) \times 10^{-6}) \sin(\gamma \times 10^{-6} + R^S/R_e) - V_{ZF}^{LOS}}{R^{S^2}} \times 10^6$$

$$\frac{-V \cos(\pi + (\theta - \alpha) \times 10^{-6}) \cos(\gamma \times 10^{-6} + R^S/R_e)}{R^S R_e} \times 10^6 \quad (D-3d)$$

$$H_{22} = \frac{\cos(\pi + (\theta - \alpha) \times 10^{-6}) \sin(\gamma \times 10^{-6} + R^S/R_e)}{R^S} \times 10^6 \quad (D-3e)$$

$$H_{23} = \frac{V \sin(\pi + (\theta - \alpha) \times 10^{-6}) \sin(\gamma \times 10^{-6} + R^S/R_e)}{R^S} \quad (D-3f)$$

APPENDIX E

ADDITIONAL MATRICES FOR ALGORITHM FIVE

This appendix gives the additional matrices required to implement algorithm five, in which the state vector is

$$[R^S \quad V_{XT}^I \quad V_{YT}^I \quad V_{ZT}^I]^T:$$

$$\underline{G}(t) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (E-1)$$

The non-zero terms of $\underline{F}[t; \hat{\underline{x}}(t/t_1)]$ are

$$F_{12} = \cos(\theta \times 10^{-6}) \cos(\gamma \times 10^{-6}) \quad (E-2a)$$

$$F_{13} = \sin(\theta \times 10^{-6}) \cos(\gamma \times 10^{-6}) \quad (E-2b)$$

$$F_{14} = \sin(\gamma \times 10^{-6}) \quad (E-2c)$$

The non-zero elements of $\underline{H}[t_1; \hat{\underline{x}}(t_1^-)]$ are

$$H_{11} = \frac{V_{YT}^I \cos(\theta \times 10^{-6}) - V_{XT}^I \sin(\theta \times 10^{-6}) - V_{YF}^{LOS}}{R^S \cos(\gamma \times 10^{-6})} \times 10^6 \quad (E-3a)$$

$$H_{12} = \frac{-\sin(\theta \times 10^{-6})}{R^S \cos(\gamma \times 10^{-6})} \quad (E-3b)$$

$$H_{13} = \frac{\cos(\theta \times 10^{-6})}{R^S \cos(\gamma \times 10^{-6})} \times 10^6 \quad (E-3c)$$

$$H_{21} = \frac{-(-V_{ZT}^I \cos(\gamma \times 10^{-6}) - [V_{XT}^I \cos(\theta \times 10^{-6}) + V_{YT}^I \sin(\theta \times 10^{-6})] \sin(\gamma \times 10^{-6}) - V_{ZF}^{LOS}}{R^S} \times 10^6 \quad (E-3d)$$

$$H_{22} = \frac{-\cos(\theta \times 10^{-6}) \sin(\gamma \times 10^{-6}) \times 10^6}{R^S} \quad (E-3e)$$

$$H_{23} = \frac{-\sin(\theta \times 10^{-6}) \sin(\gamma \times 10^{-6}) \times 10^6}{R^S} \quad (E-3f)$$

$$H_{24} = \frac{-\cos(\gamma \times 10^{-6}) \times 10^6}{R^S} \quad (E-3g)$$

When noise is added to the range channel as suggested at the end of Section 3.3.5, Equation (E-1) becomes

$$\underline{G}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (E-4)$$

All other matrices given here remain the same.

APPENDIX F
ADDITIONAL MATRICES FOR ALGORITHM SIX

This appendix contains the additional matrices needed to implement algorithm six, in which the state vector is

$$[R^S \quad V_{XT}^{LOS} \quad V_{YT}^{LOS} \quad V_{ZT}^{LOS}]^T:$$

$$\underline{G}(t) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (F-1)$$

There is only one non-zero value in $\underline{F}[t; \hat{\underline{x}}(t/t_1)]$, given by

$$F_{12} = 1 \quad (F-2)$$

The non-zero elements of $\underline{H}[t_1; \hat{\underline{x}}(t_1^-)]$ are

$$H_{11} = \frac{-(V_{YT}^{LOS} - V_{YF}^{LOS}) \times 10^6}{R^S \cos(\gamma \times 10^{-6})} \quad (F-3a)$$

$$H_{13} = \frac{10^6}{R^S} \quad (F-3b)$$

$$H_{21} = \frac{-(V_{ZT}^{LOS} - V_{ZF}^{LOS}) \times 10^6}{R^S} \quad (F-3c)$$

$$H_{24} = \frac{10^6}{R^S} \quad (F-3d)$$

APPENDIX G

GRAPHICAL RESULTS FOR ALGORITHM ONE

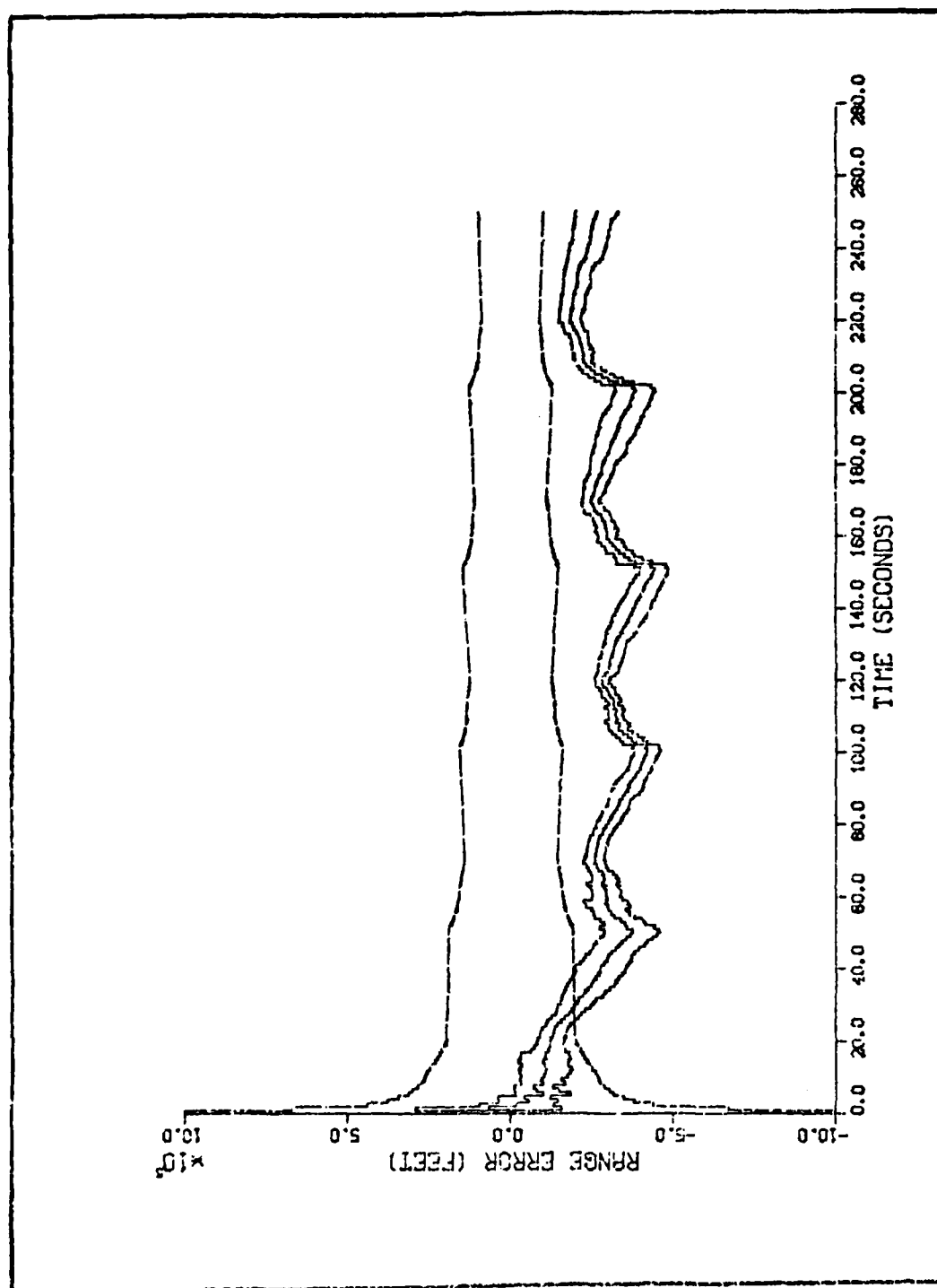


Figure G-1. Algorithm 1, Range Error Versus Time

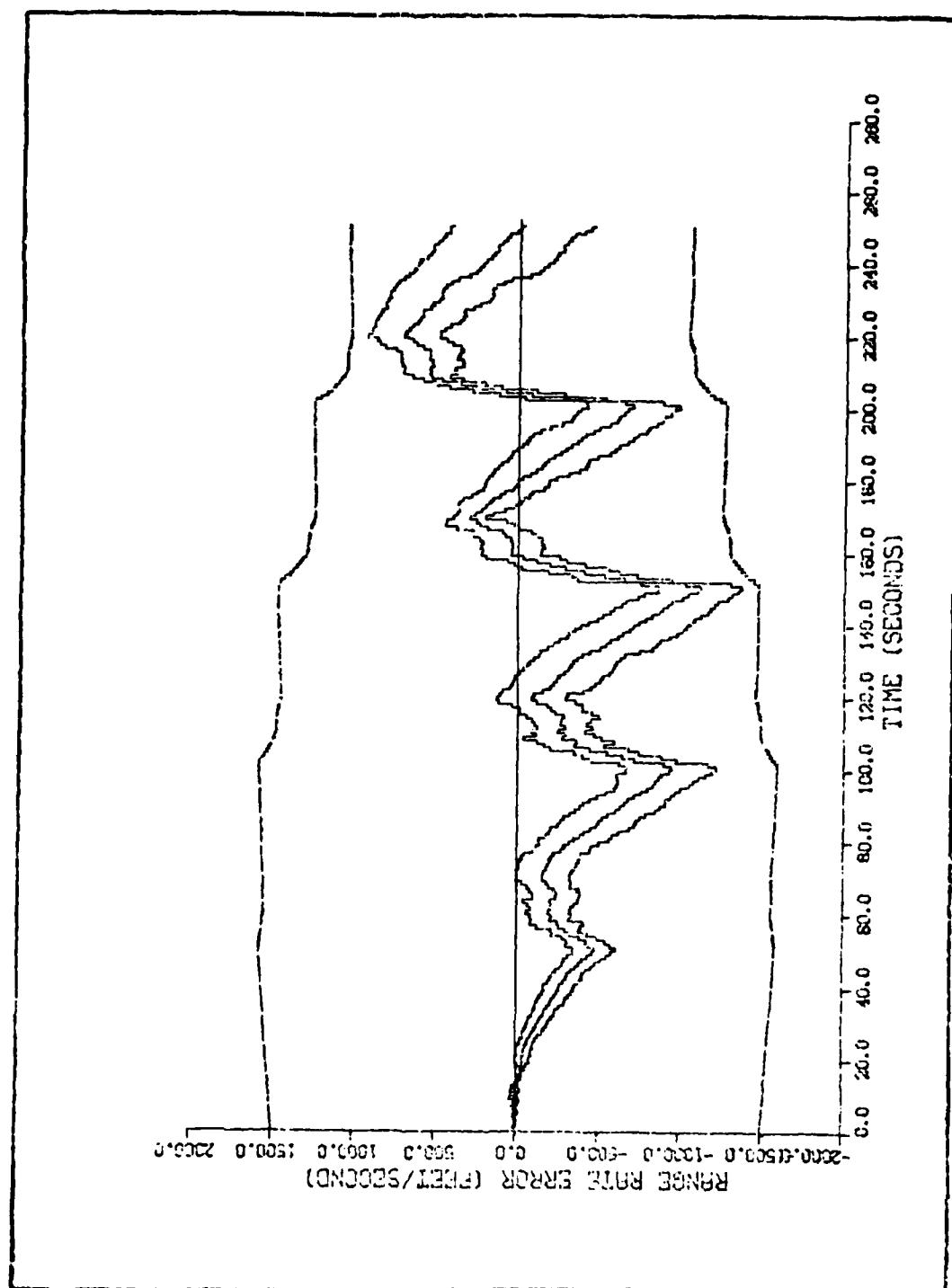


Figure G-2. Algorithm 1, Range Rate Versus Time

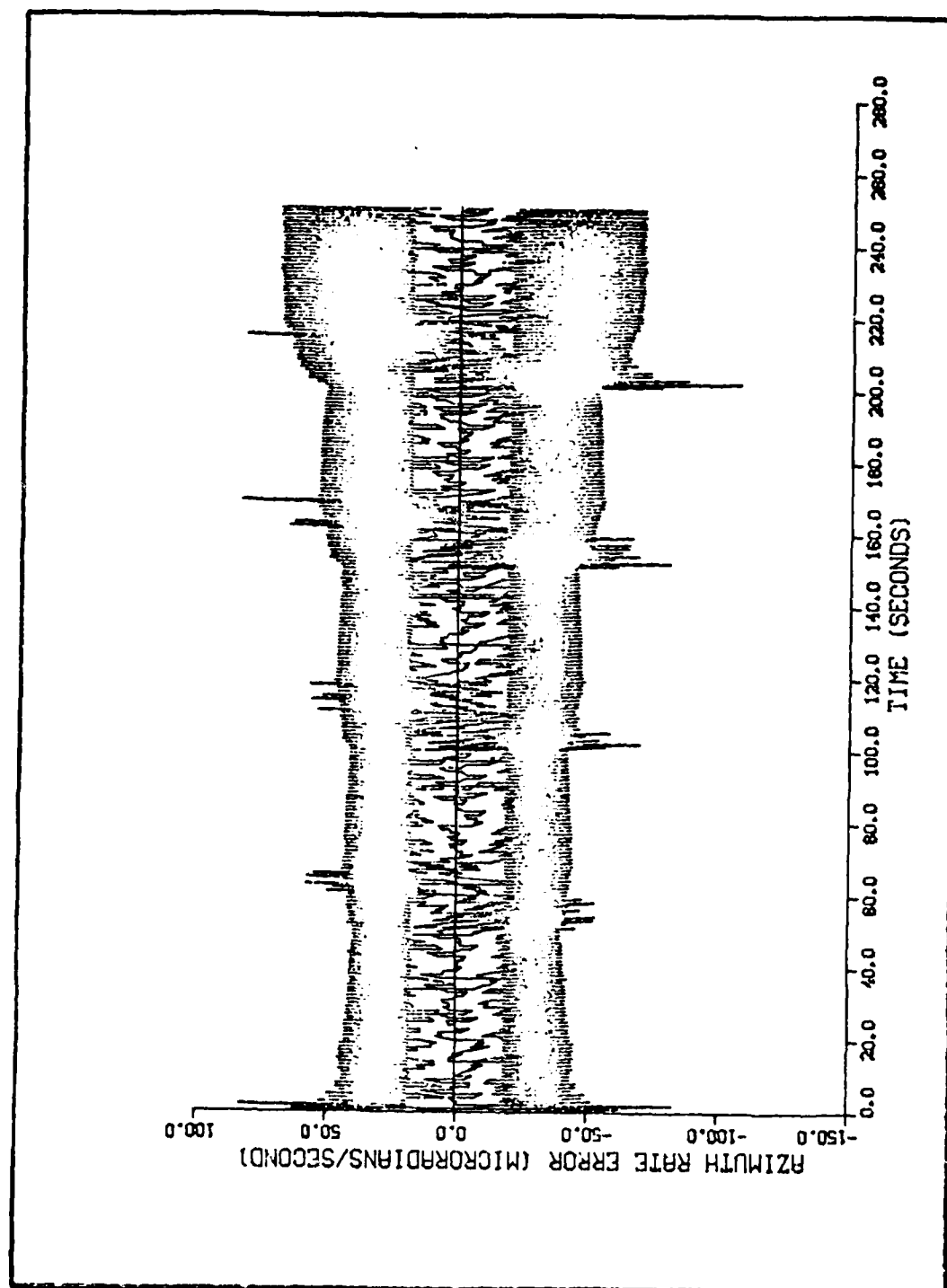


Figure G-3. Algorithm 1, Azimuth Rate Error Versus Time

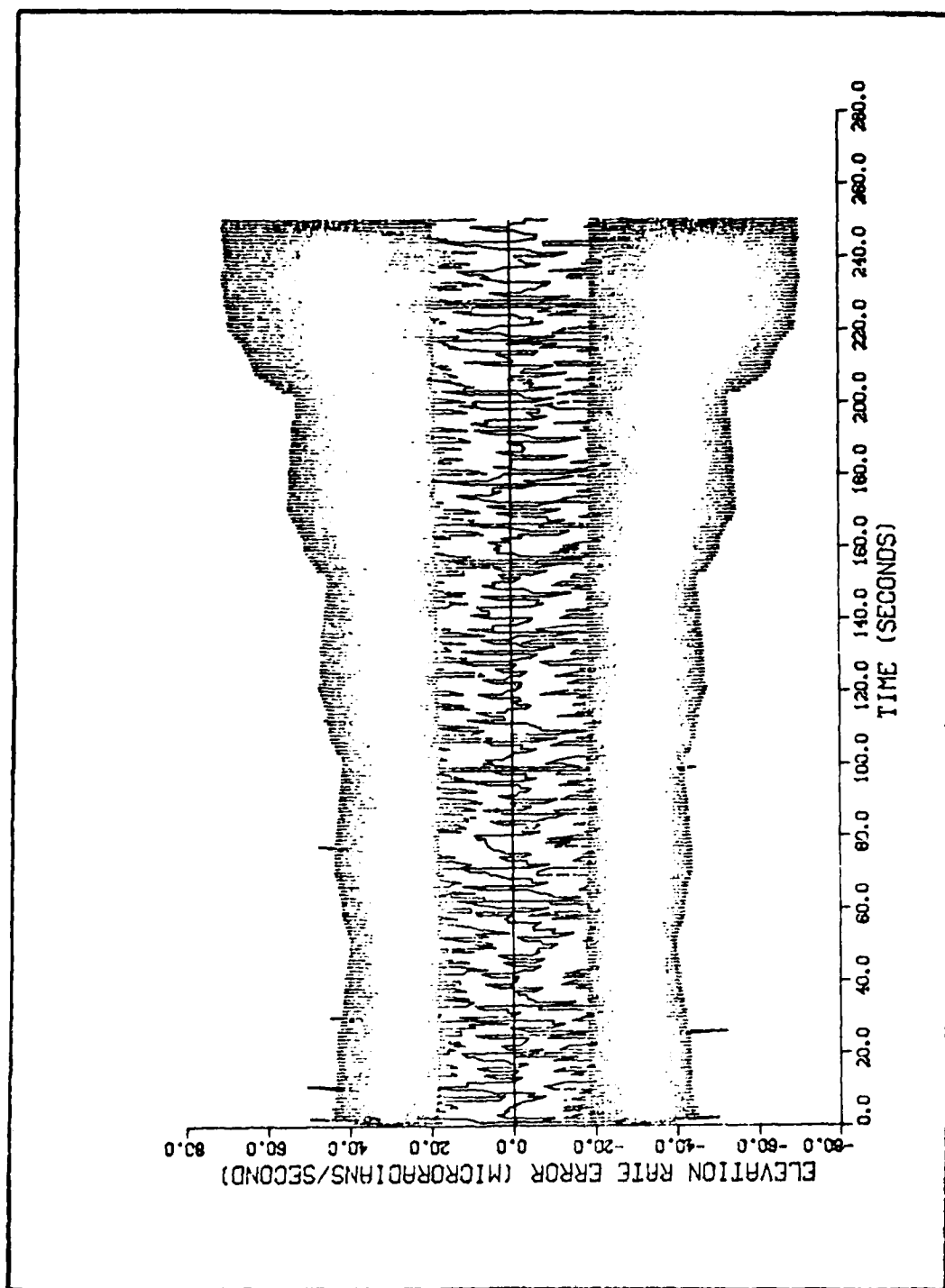


Figure G-4. Algorithm 1, Evaluation Rate Error Versus Time

APPENDIX H
GRAPHICAL RESULTS FOR ALGORITHM TWO

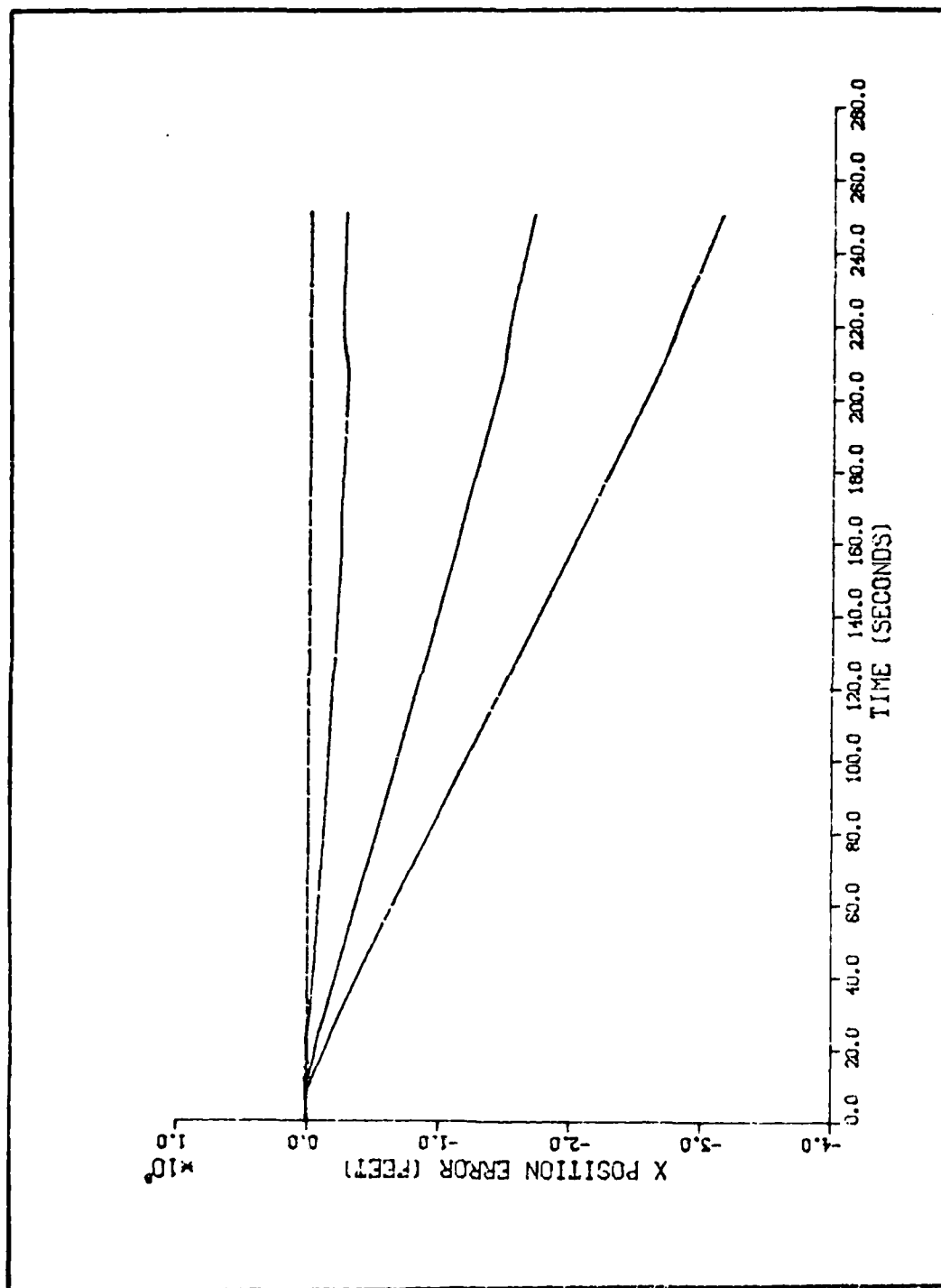


Figure H-1. Algorithm 2, Position Error in the X Direction Versus Time

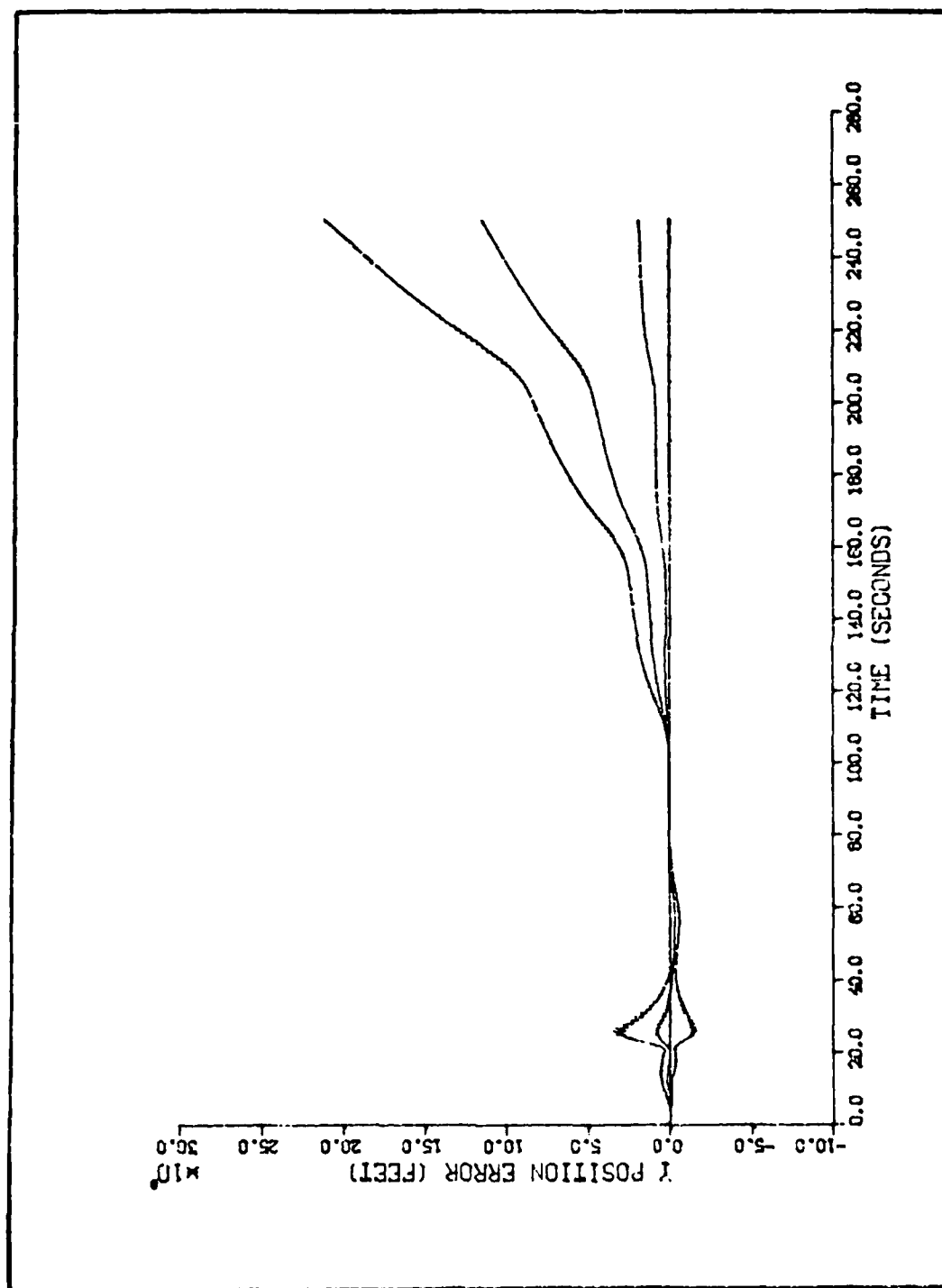


Figure H-2. Algorithm 2, Position Error in the Y Direction Versus Time

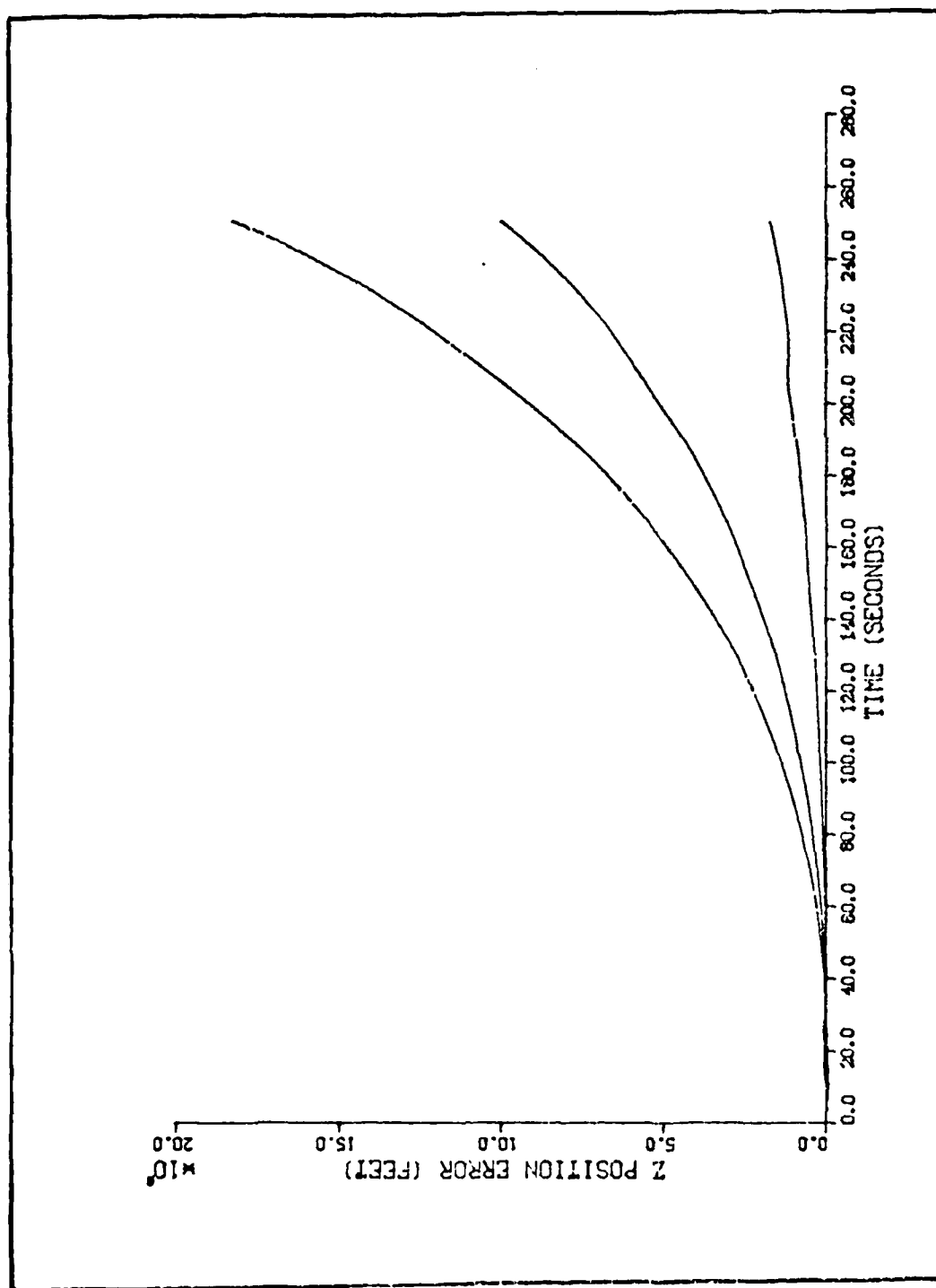


Figure H-3. Algorithm 2, Position Error in the Z Direction Versus Time

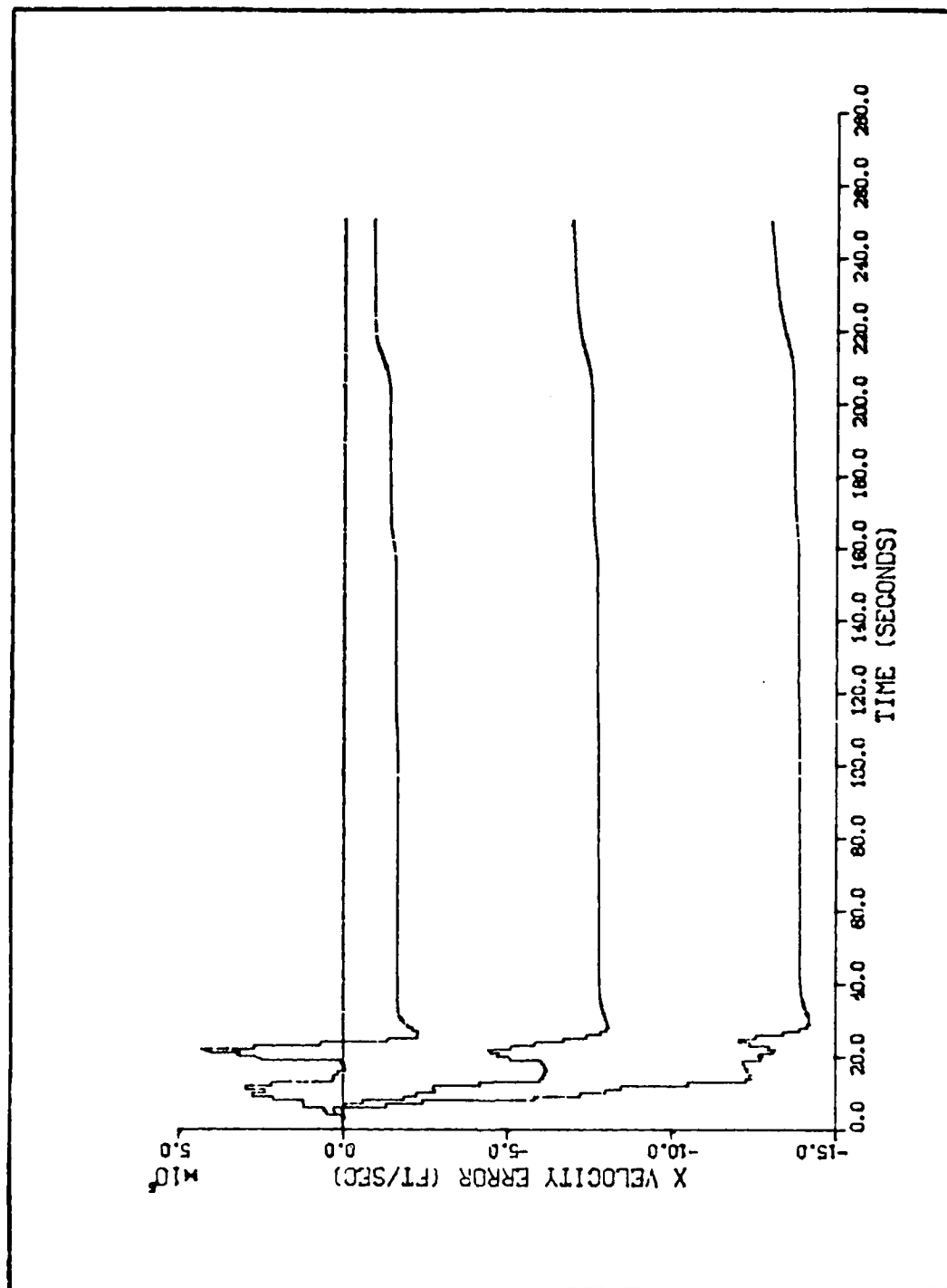


Figure H-4. Algorithm 2, Velocity Error in the X Direction Versus Time

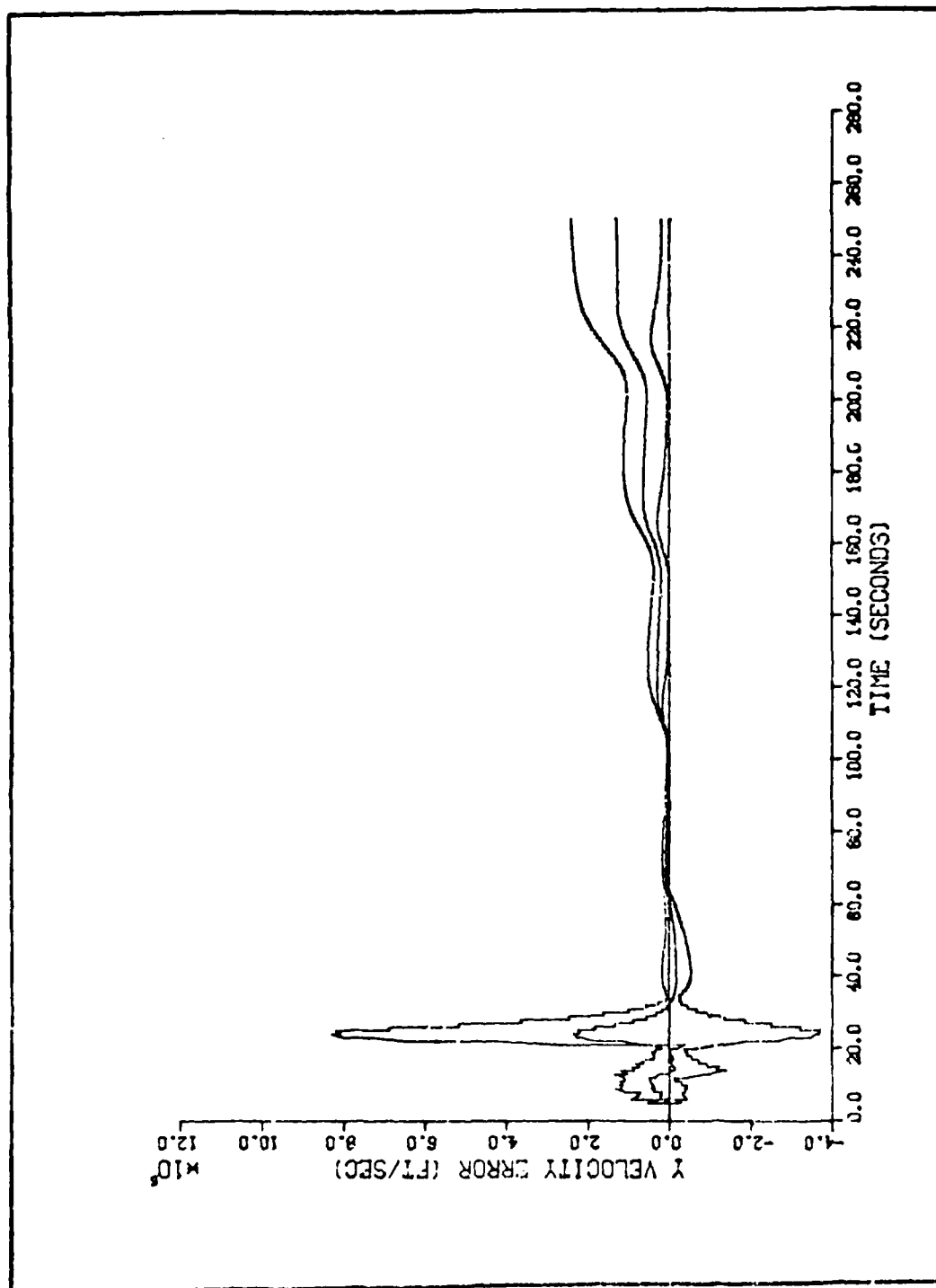


Figure H-5. Algorithm 2, Velocity Error in the Y Direction Versus Time

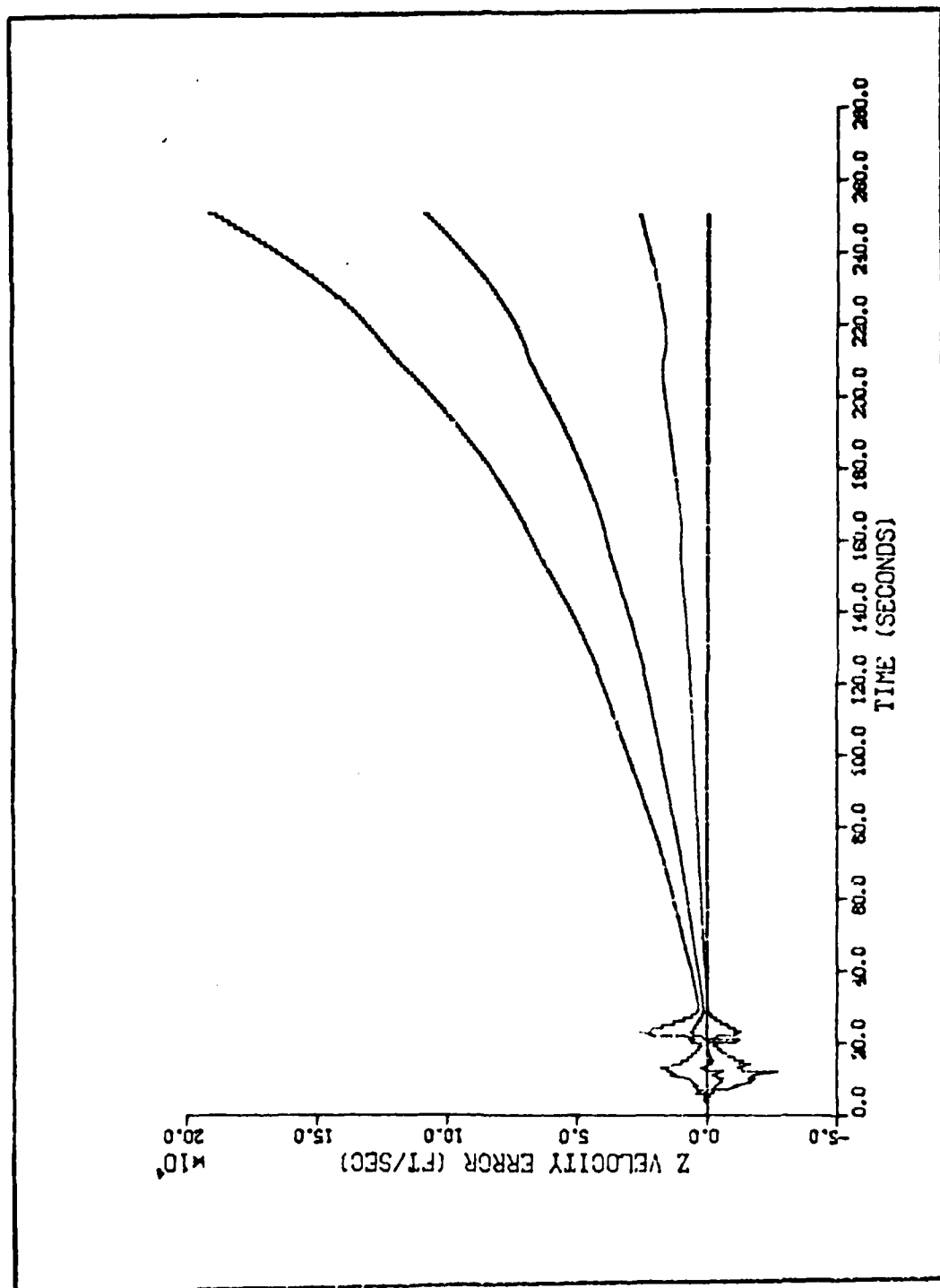


Figure H-6. Algorithm 2, Velocity Error in the Z Direction Versus Time

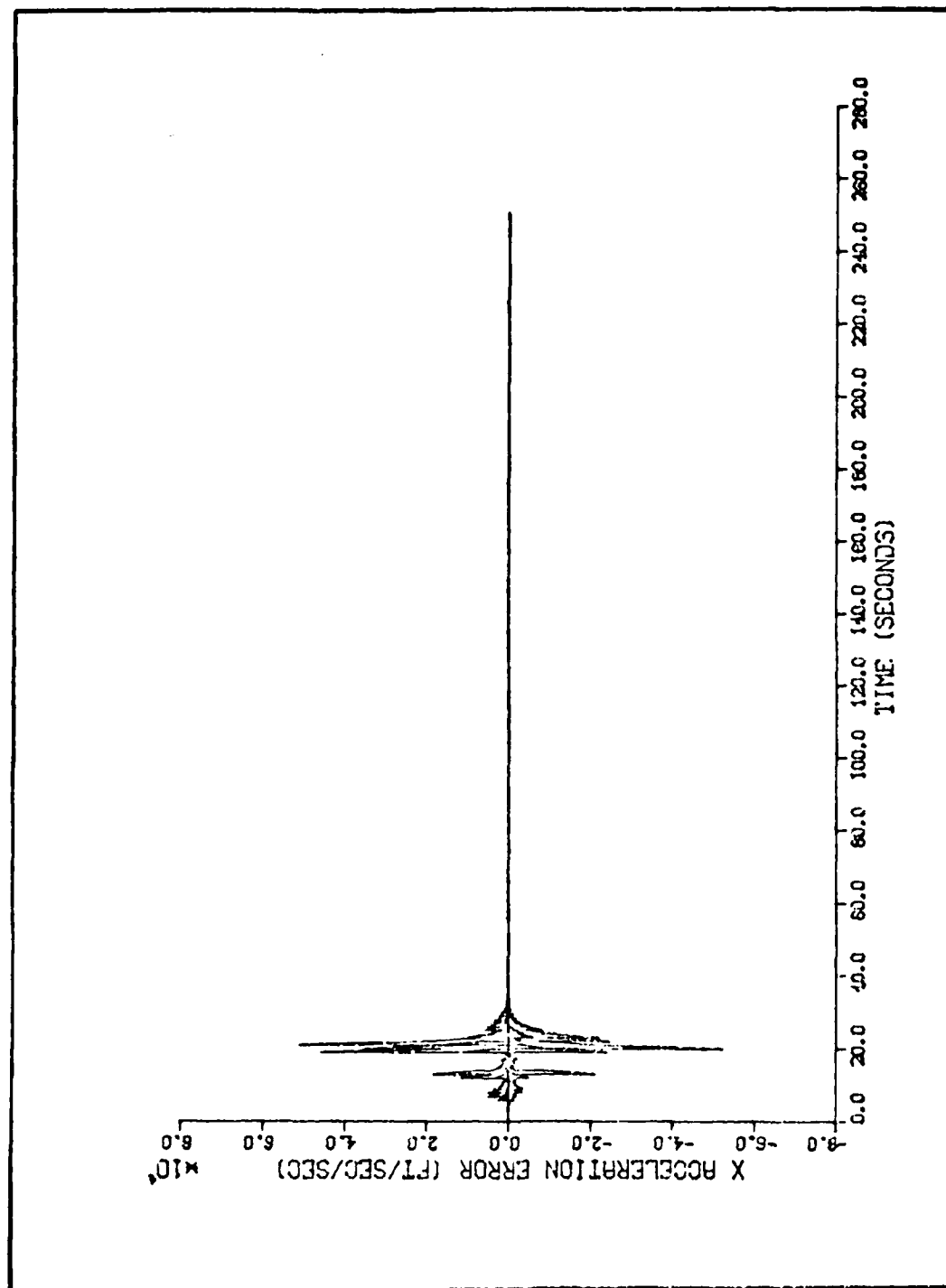


Figure H-7. Algorithm 2, Acceleration Error in the X Direction Versus Time

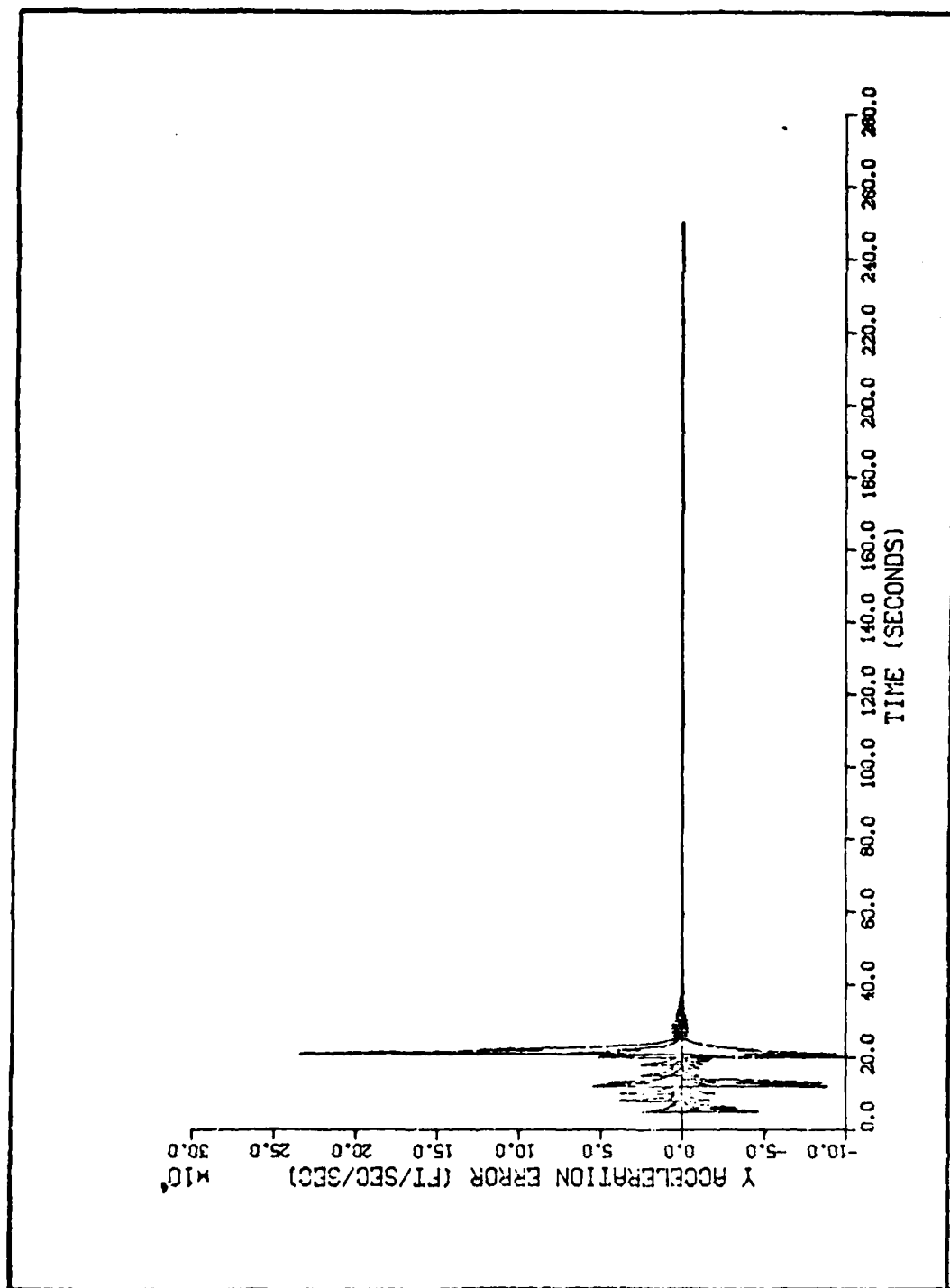


Figure H-8. Algorithm 2, Acceleration Error in the Y Direction Versus Time

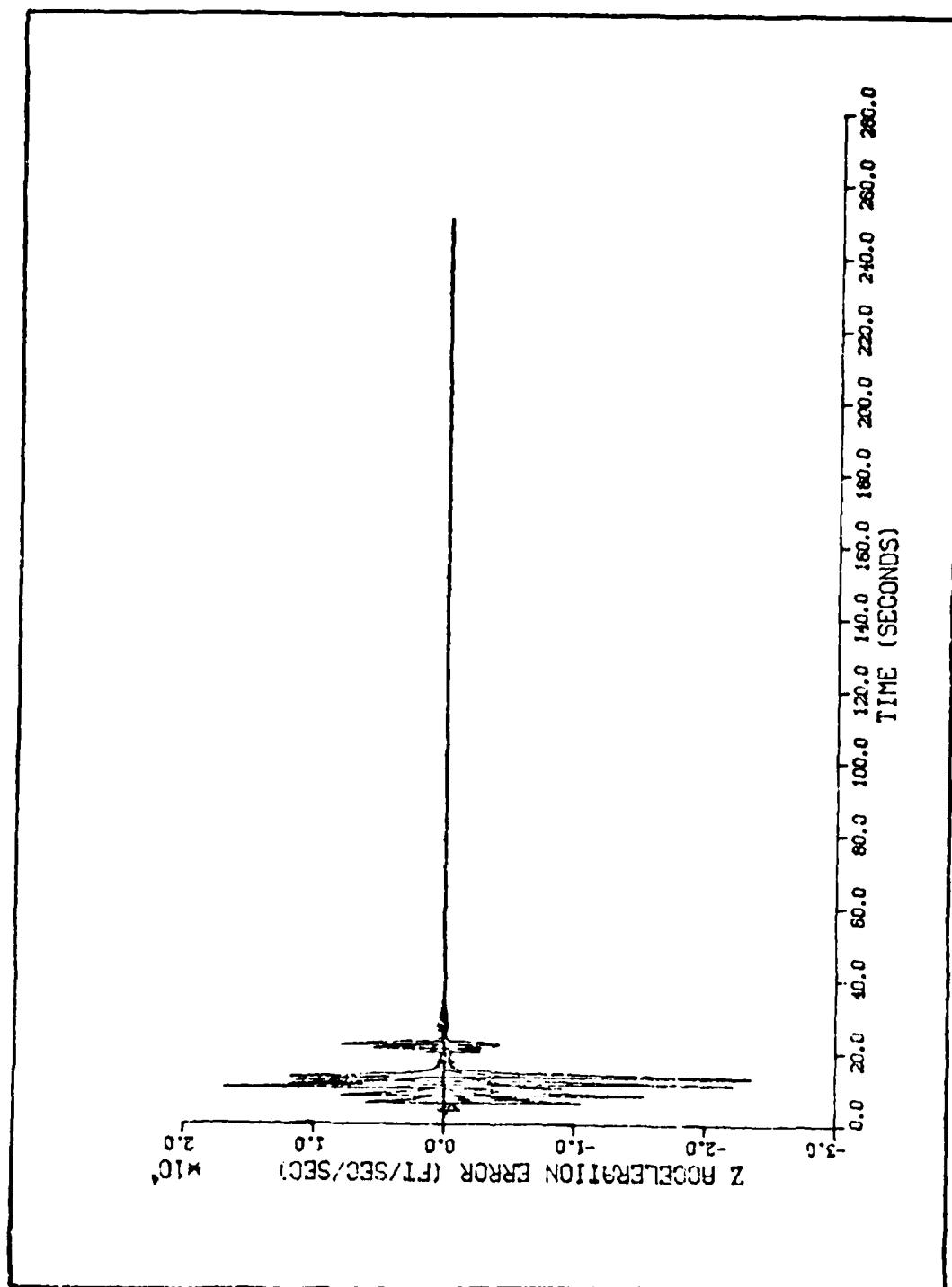


Figure H-9. Algorithm 2, Acceleration Error in the Z Direction Versus Time

APPENDIX I
GRAPHICAL RESULTS FOR ALGORITHM THREE

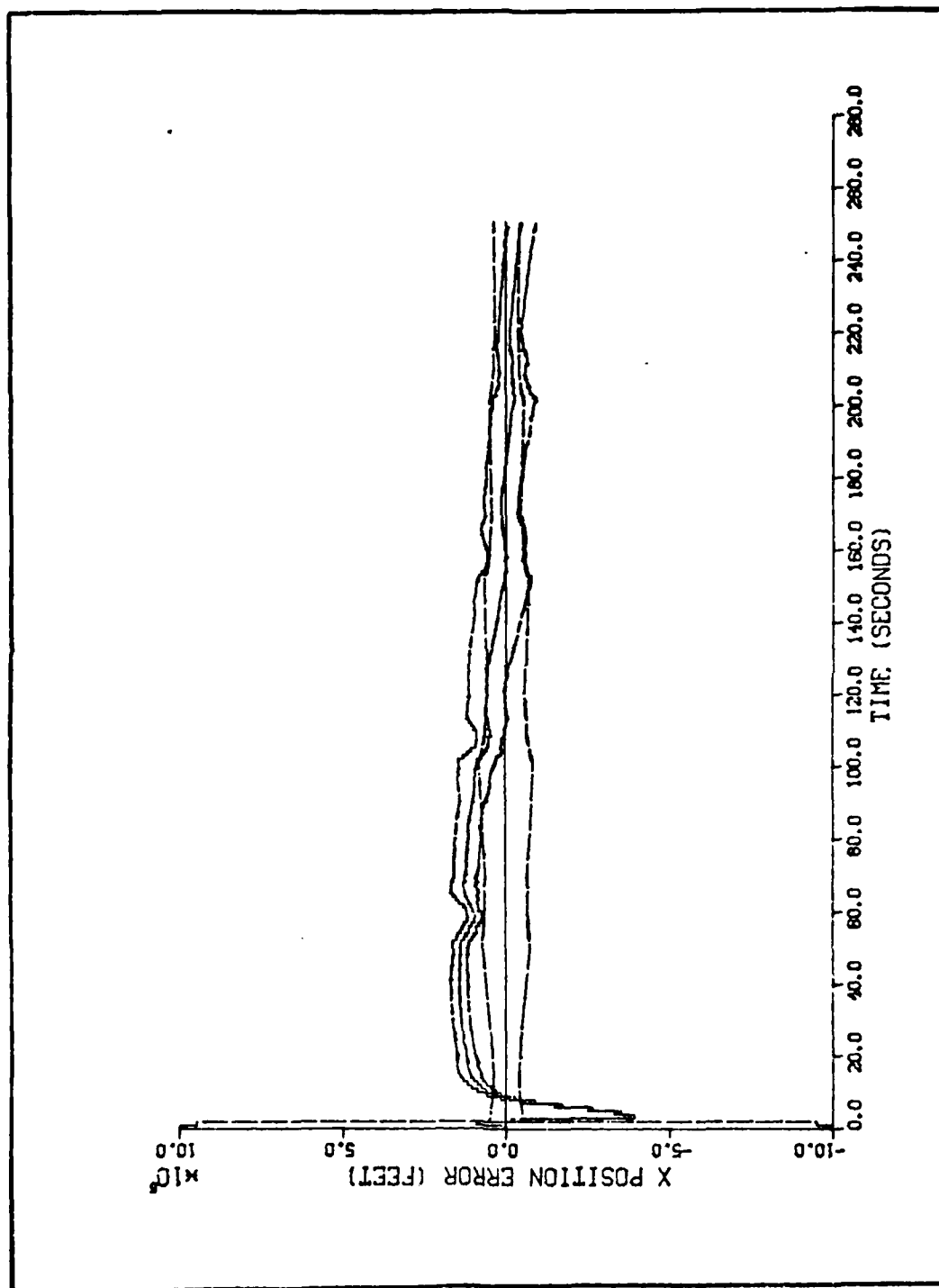


Figure I-1. Algorithm 3, Position Error in the X Direction Versus Time

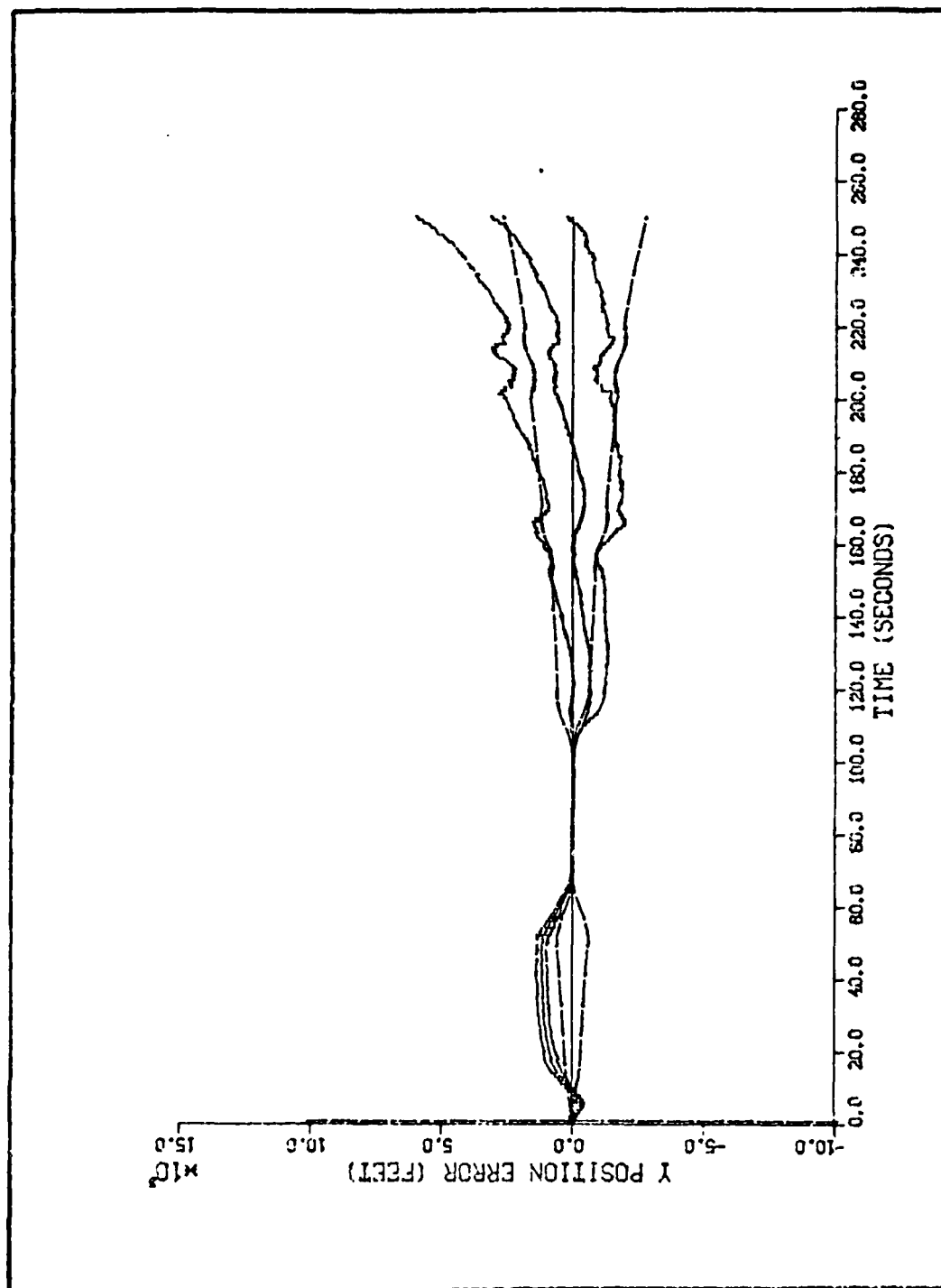


Figure I-2. Algorithm 3, Position Error in the Y Direction Versus Time

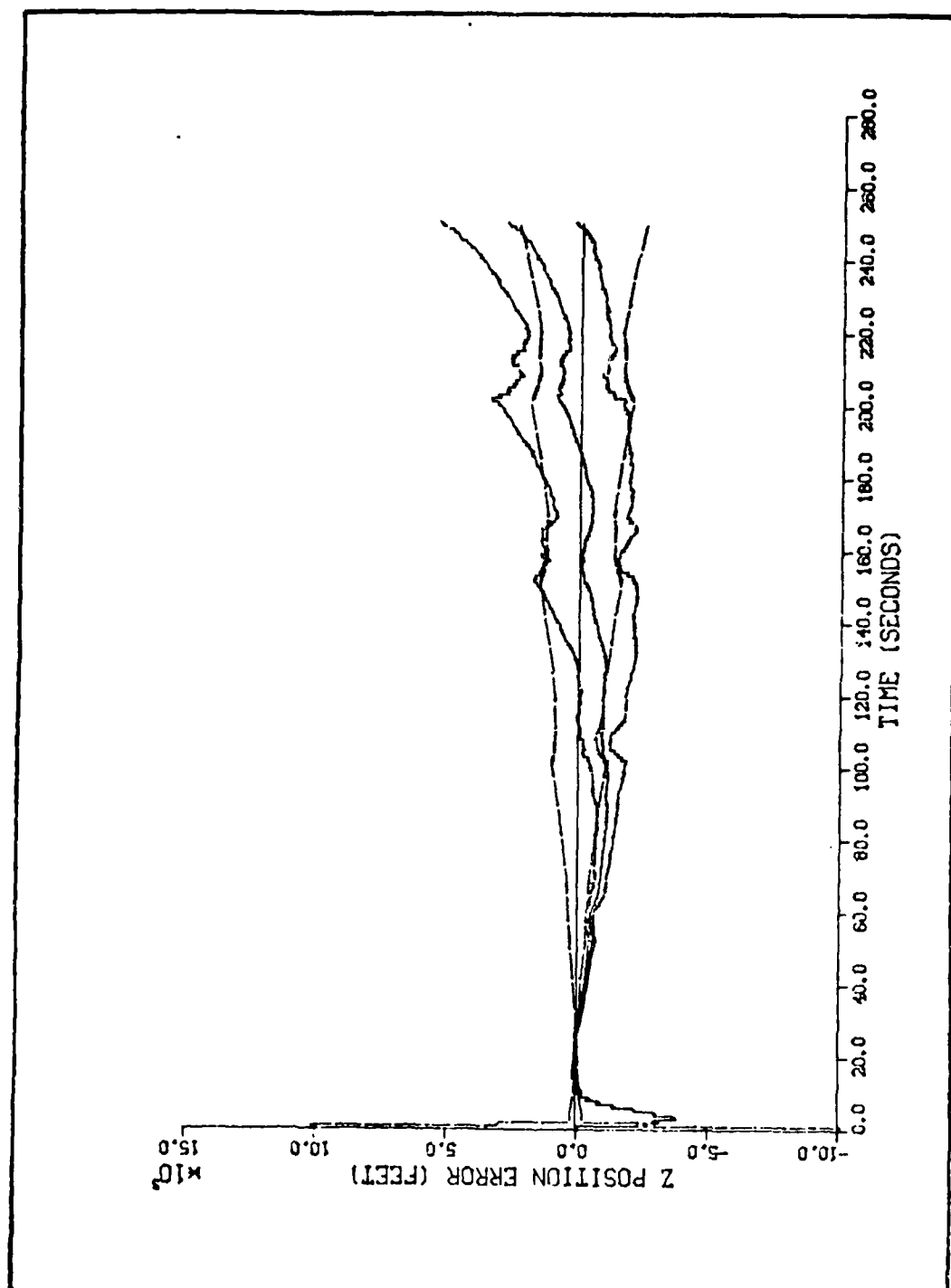


Figure I-3. Algorithm 3, Position Error in the Z Direction Versus Time

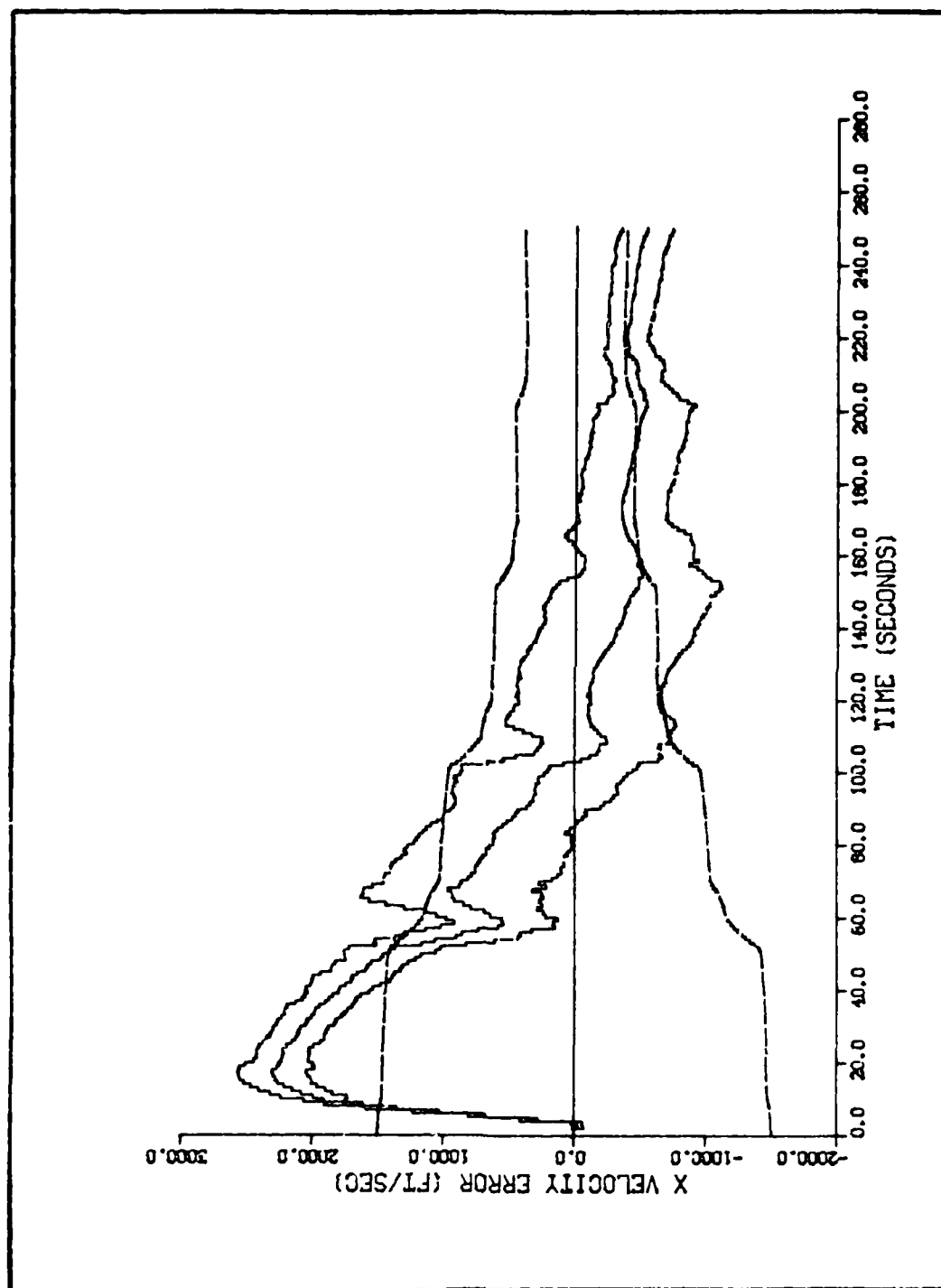


Figure I-4. Algorithm 3, Velocity Error in the X Direction Versus Time

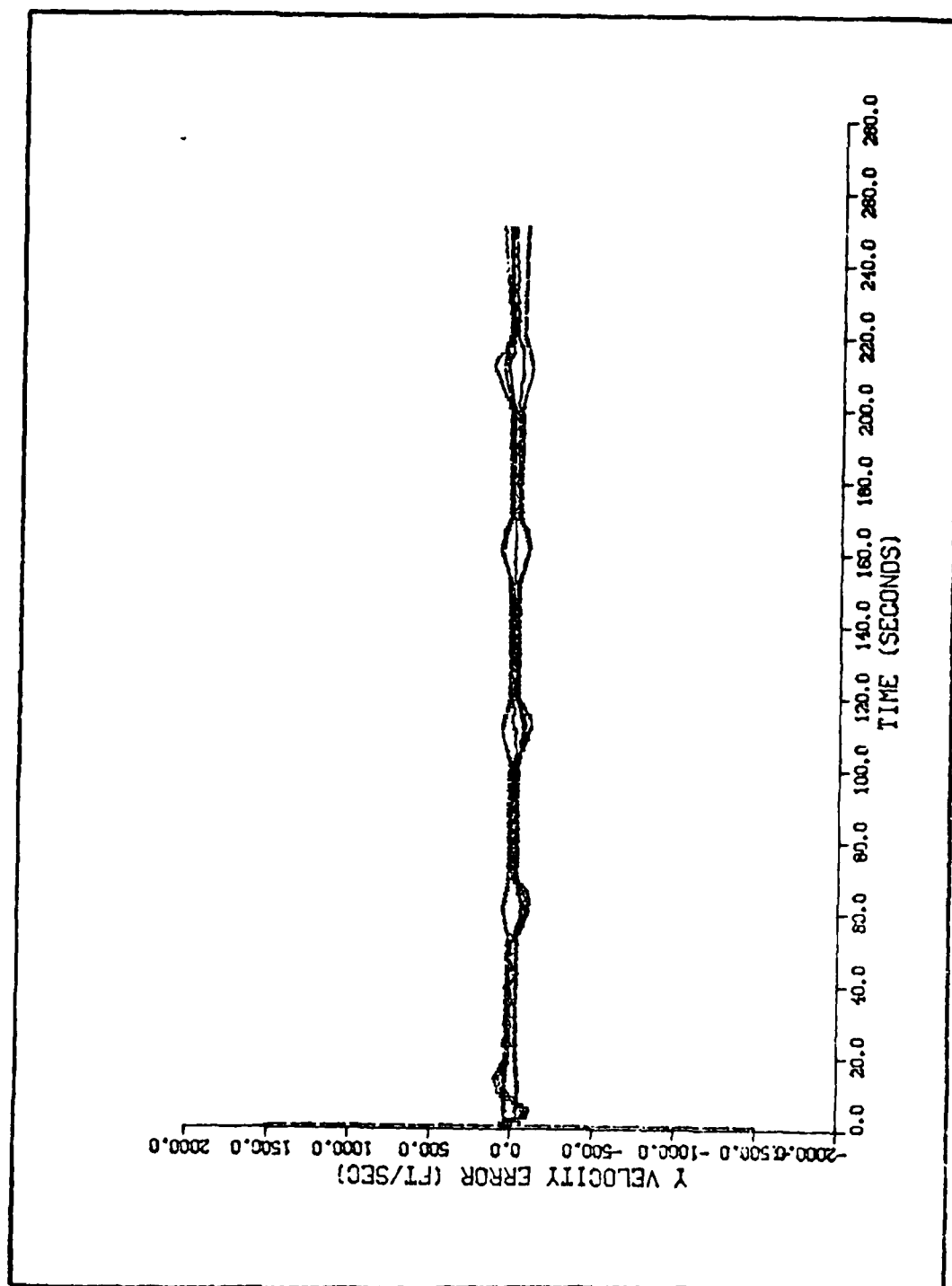


Figure I-5. Algorithm 3, Velocity Error in the Y Direction Versus Time

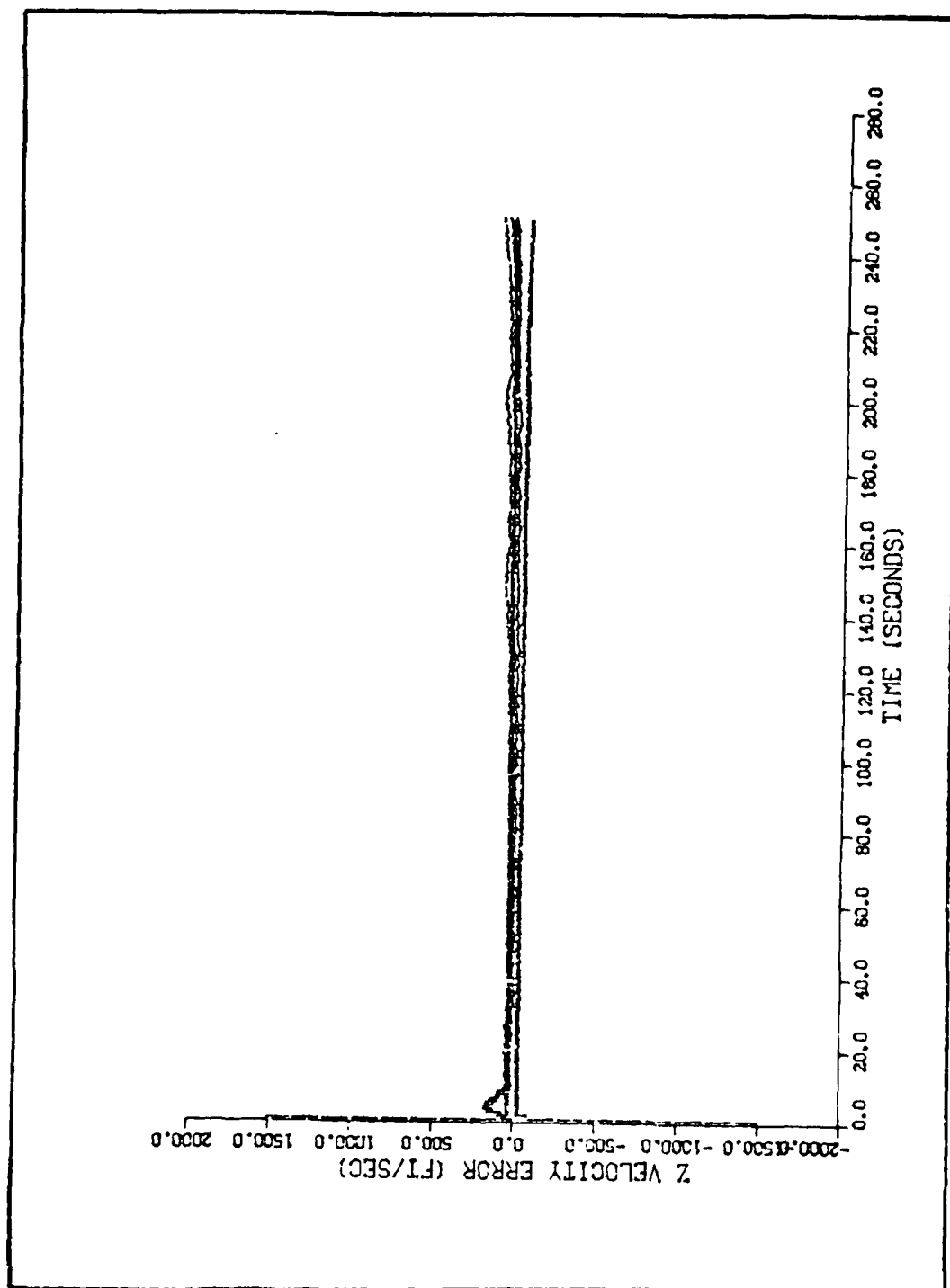


Figure I-6. Algorithm 3, Velocity Error in the Z Direction Versus Time

APPENDIX J

GRAPHICAL RESULTS FOR ALGORITHM FOUR

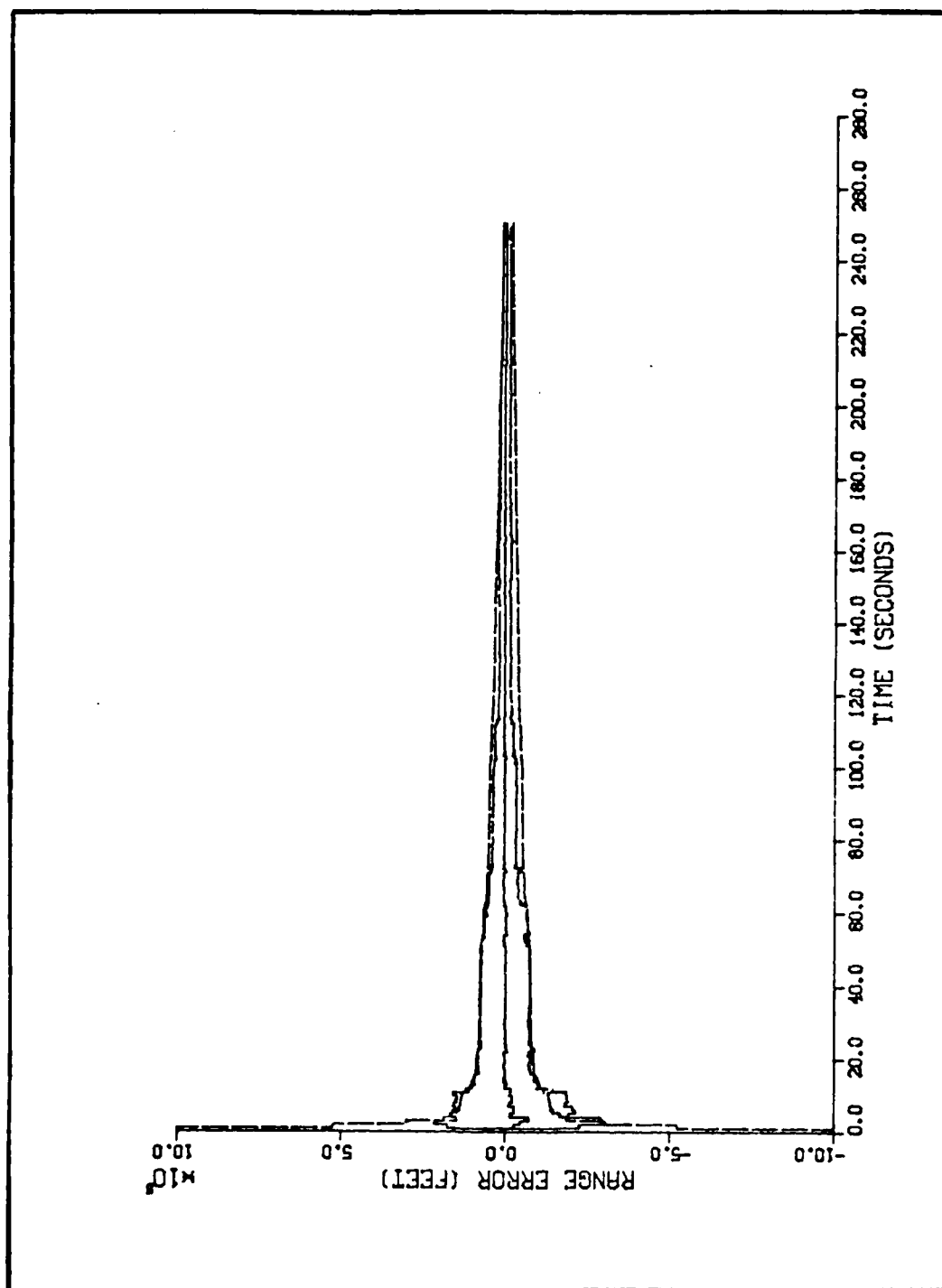


Figure J-1. Algorithm 4, Range Error Versus Time, Non-Maneuvering Target Model

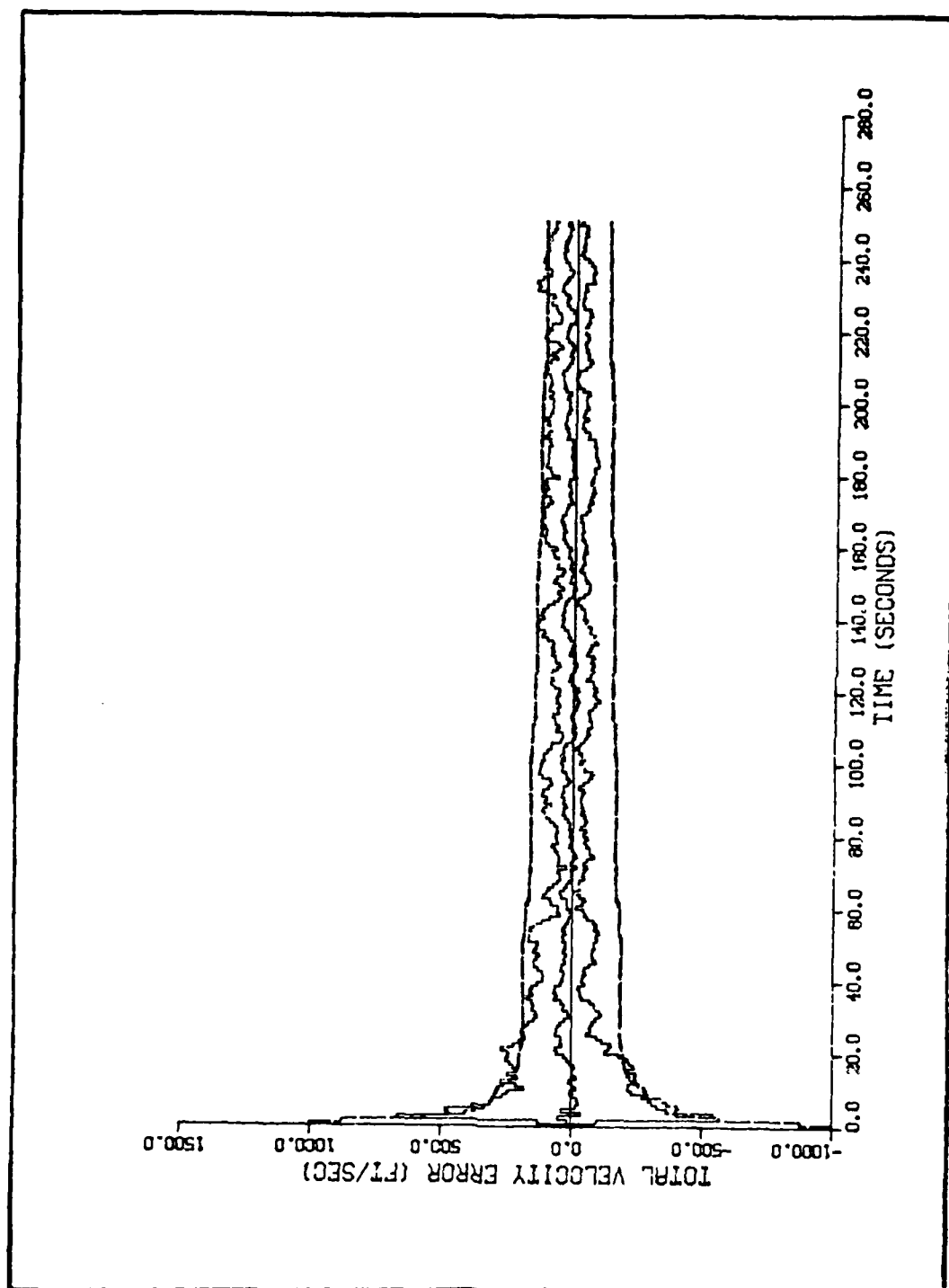


Figure J-2. Algorithm 4, Total Velocity Error Versus Time, Non-Maneuvering Target Model

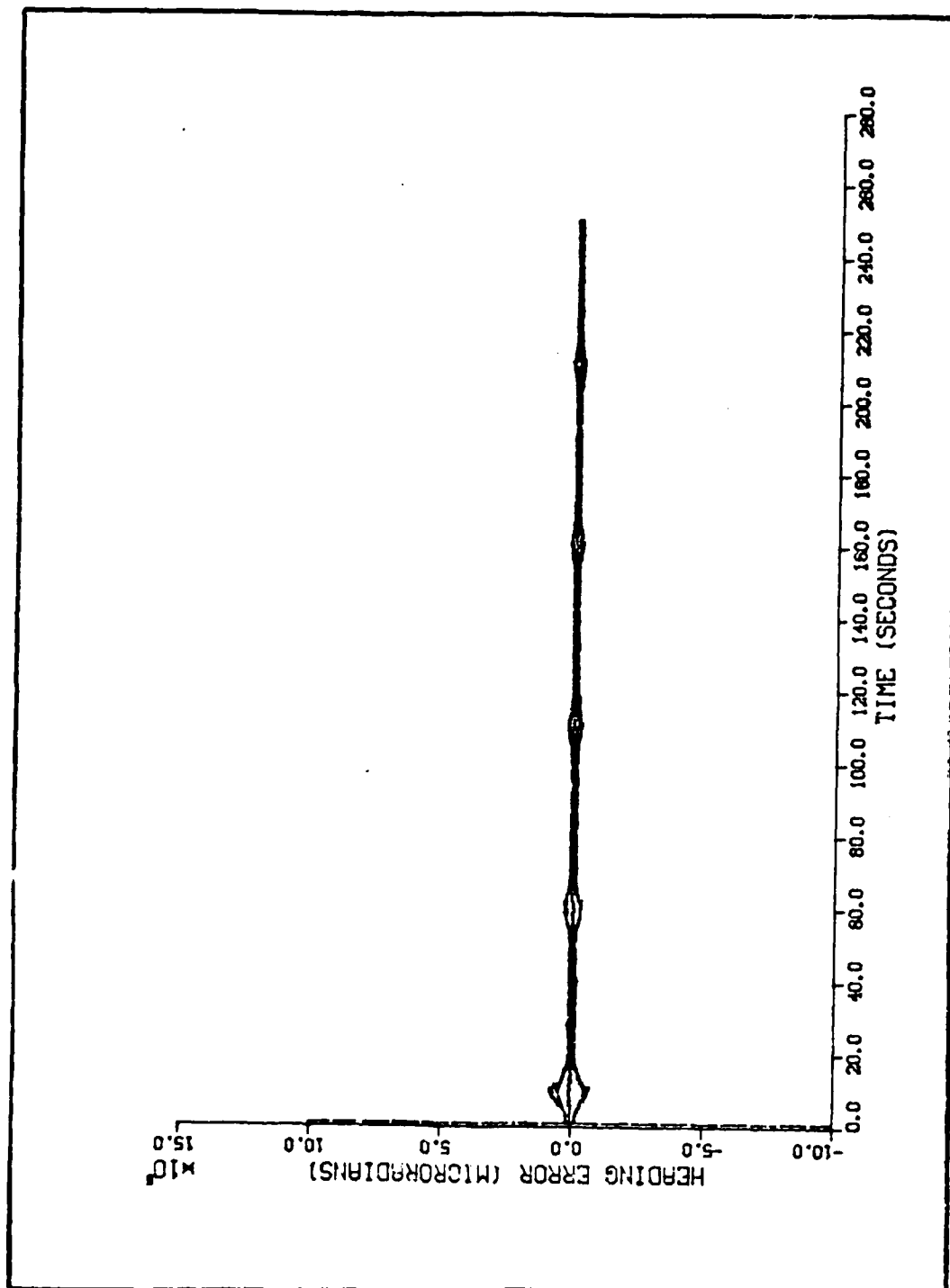


Figure J-3. Algorithm 4, Heading Error Versus Time, Non-Maneuvering Target Model

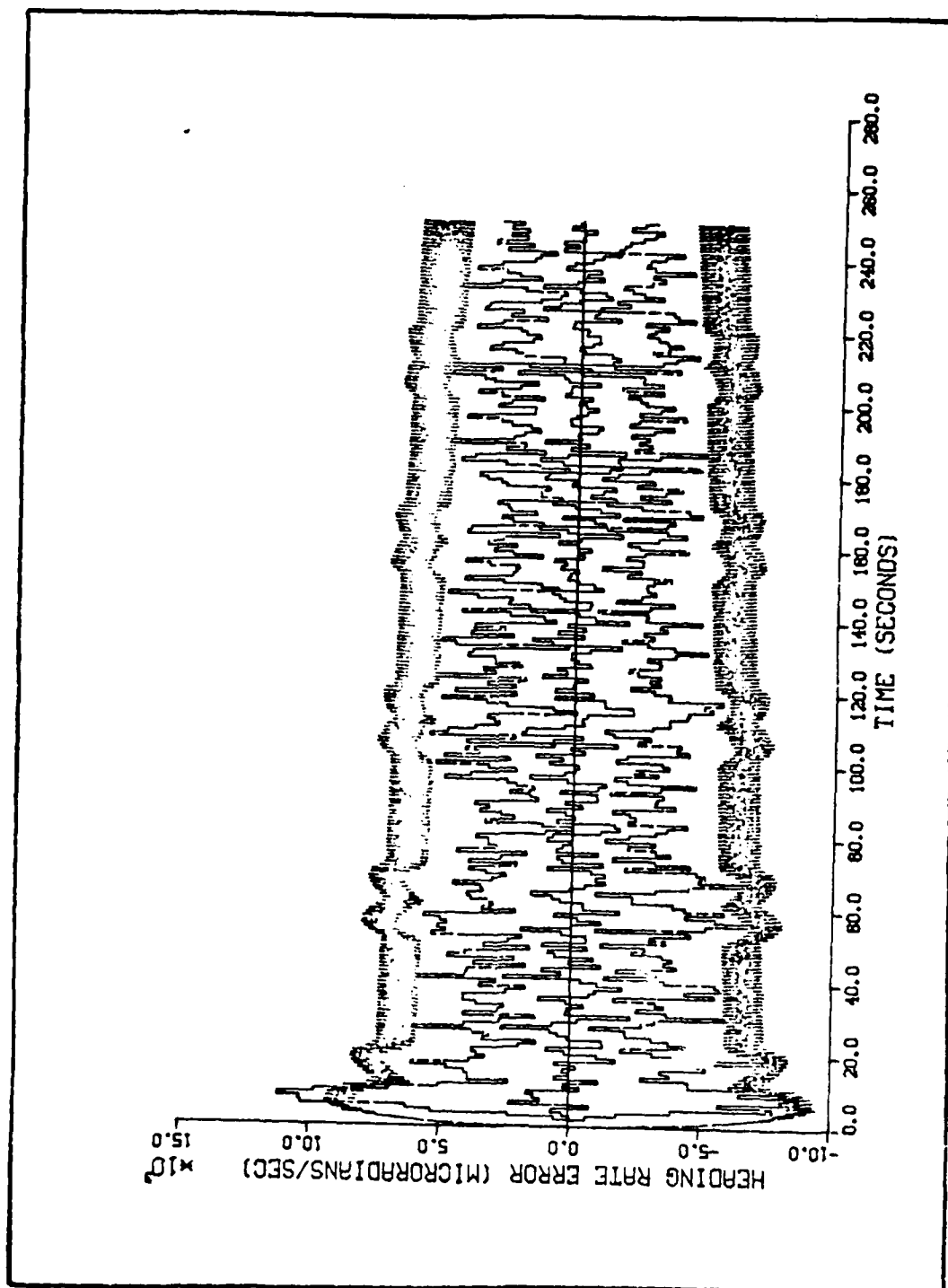


Figure J-4. Algorithm 4, Heading Rate Error Versus Time, Non-Maneuvering Target Model

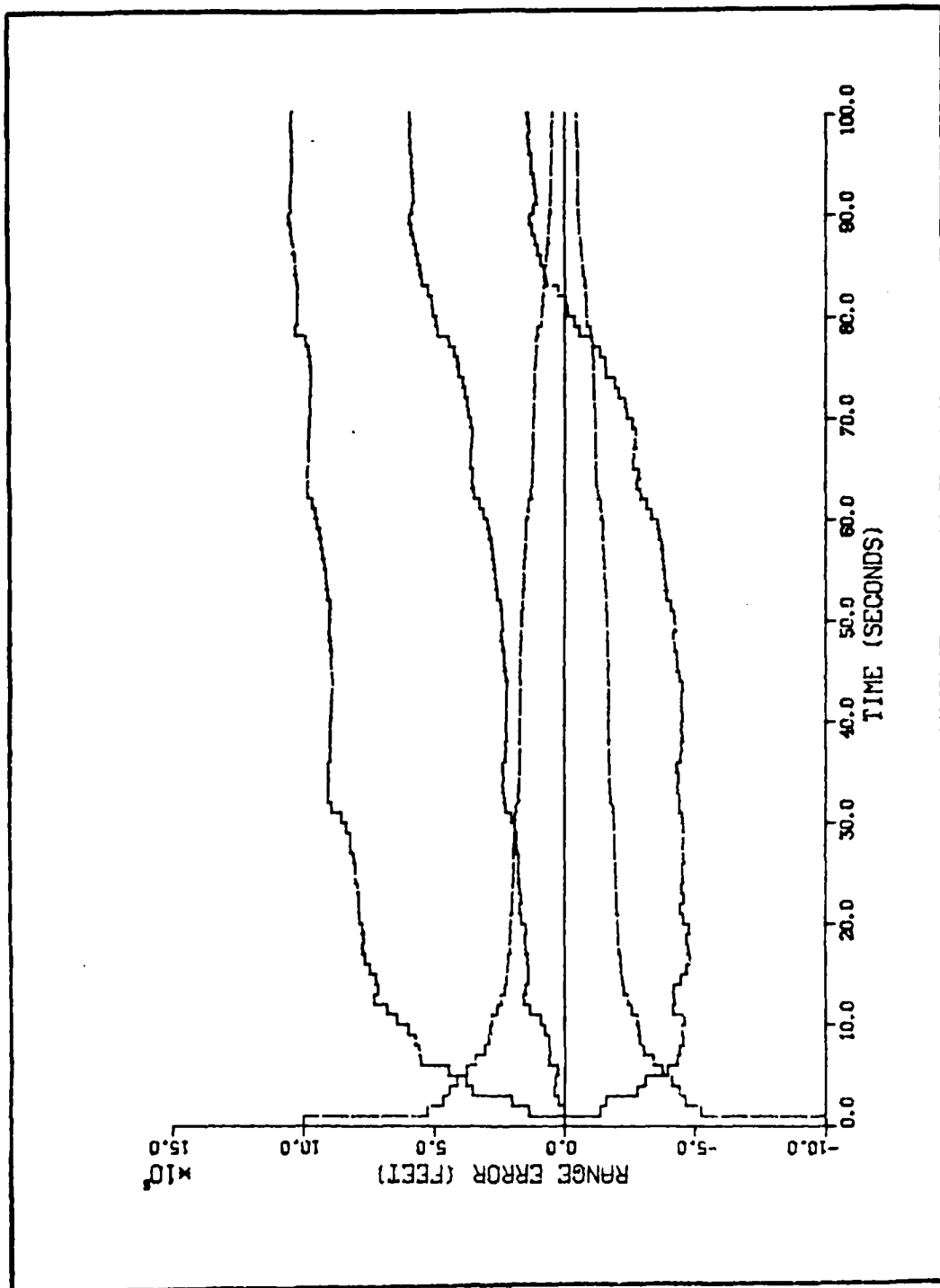


Figure J-5. Algorithm 4, Range Error Versus Time, Maneuvering Target Model

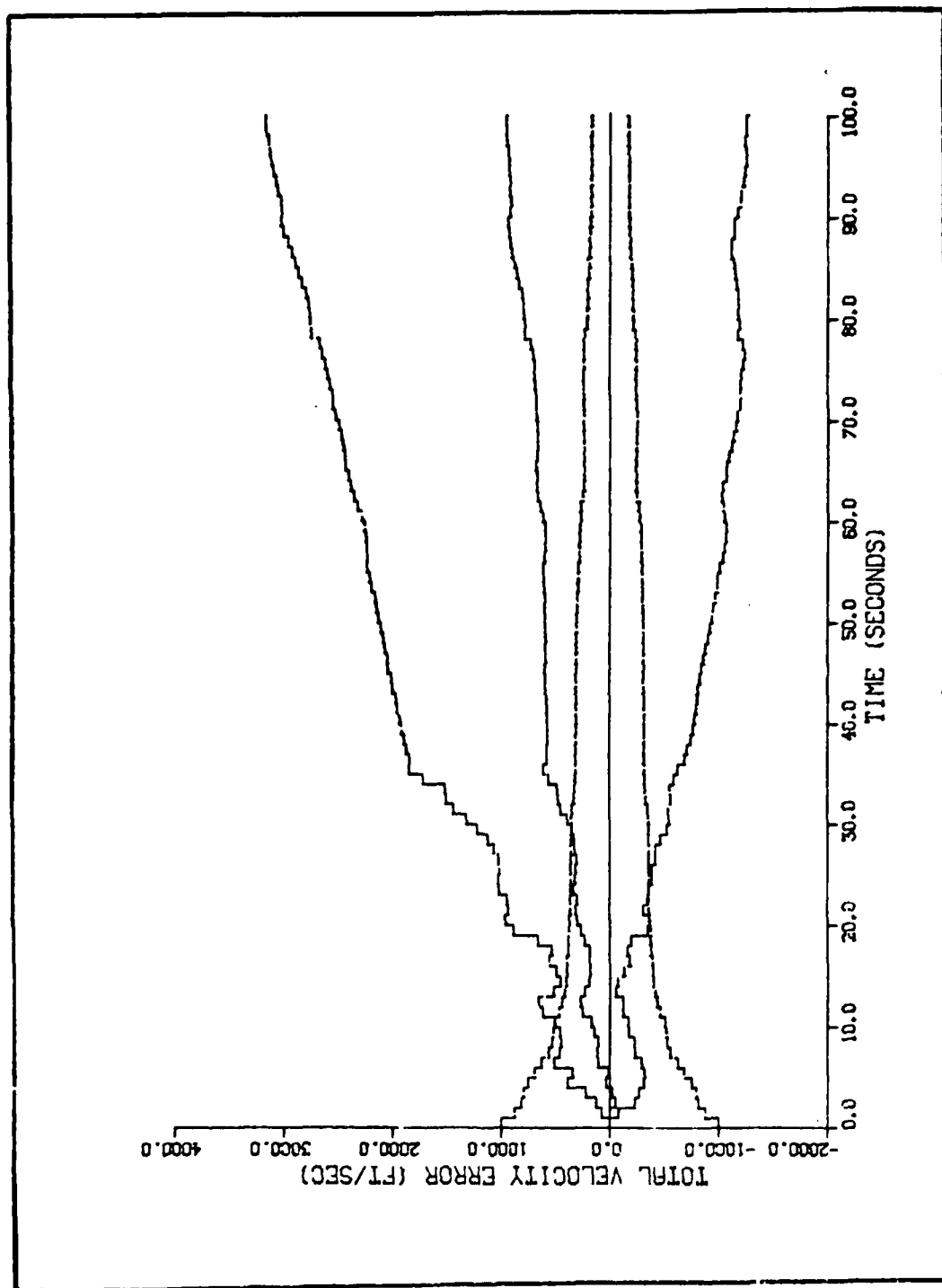


Figure J-6. Algorithm 4, Total Velocity Error Versus Time, Maneuvering Target Model

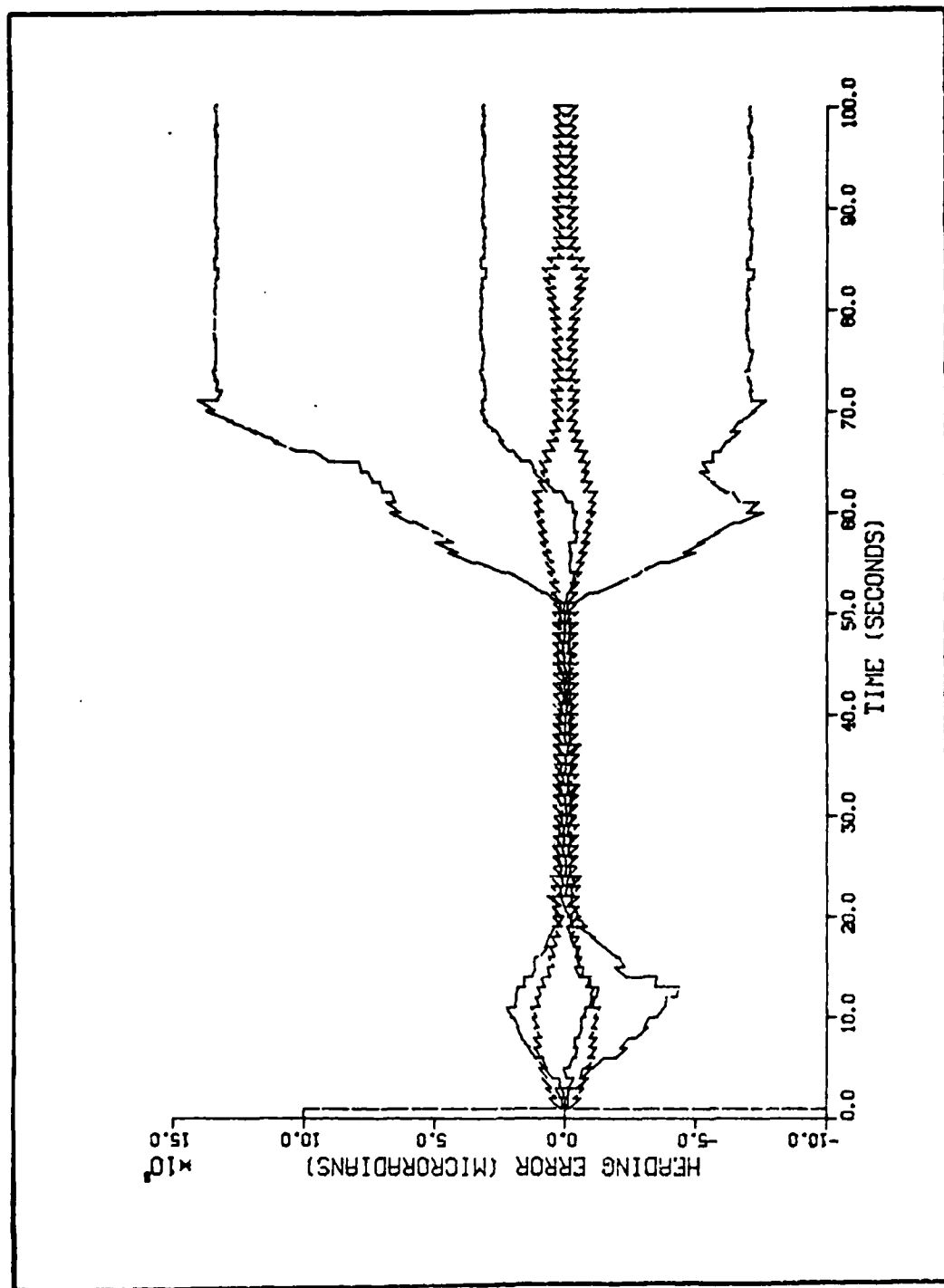


Figure J-7. Algorithm 4, Heading Error Versus Time, Maneuvering Target Model

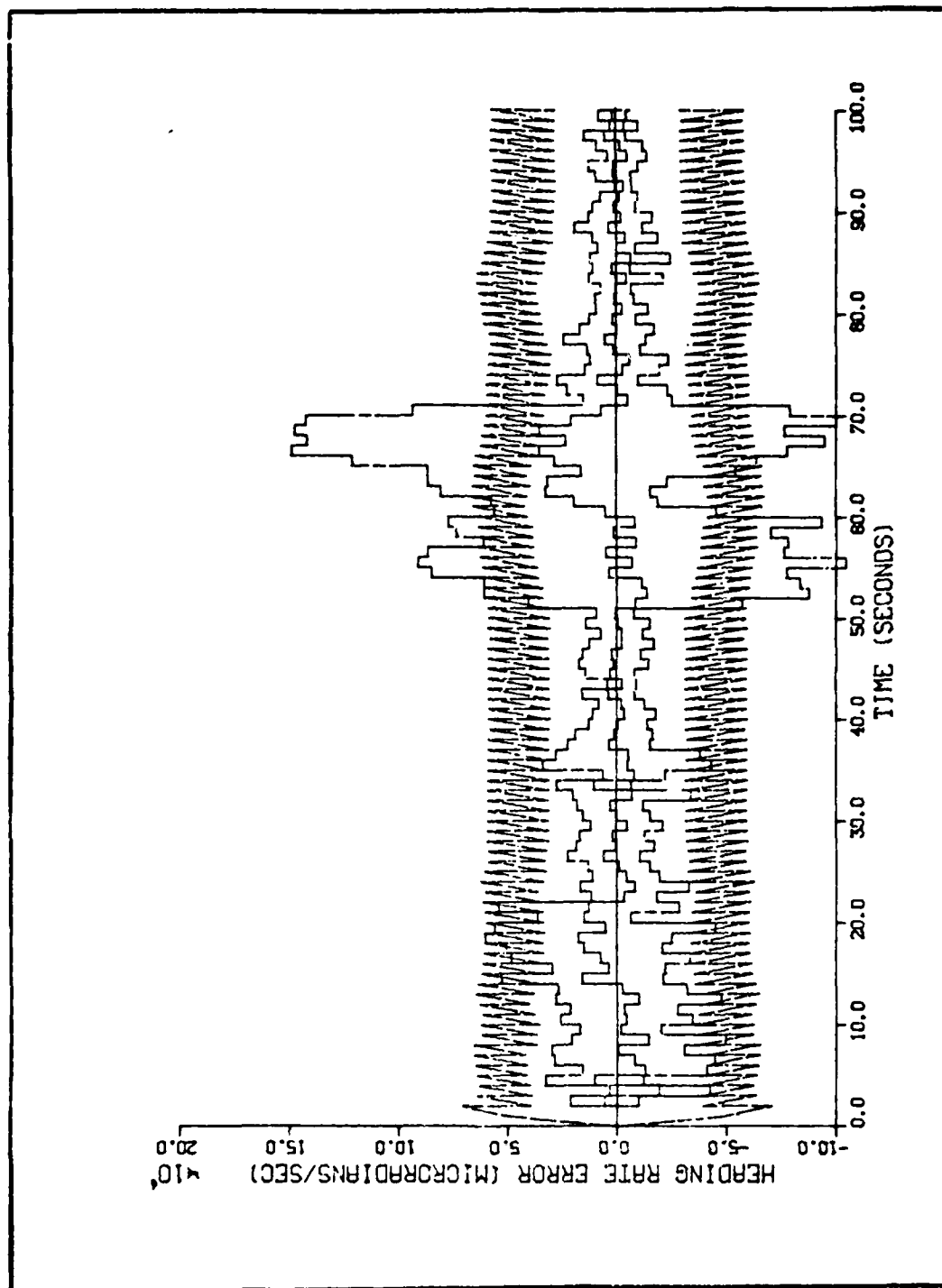


Figure J-8. Algorithm 4, Heading Rate Error Versus Time, Maneuvering Target Model

APPENDIX K

GRAPHICAL RESULTS OF ADDING PSEUDONOISE TO
THE RANGE CHANNEL OF ALGORITHM FIVE

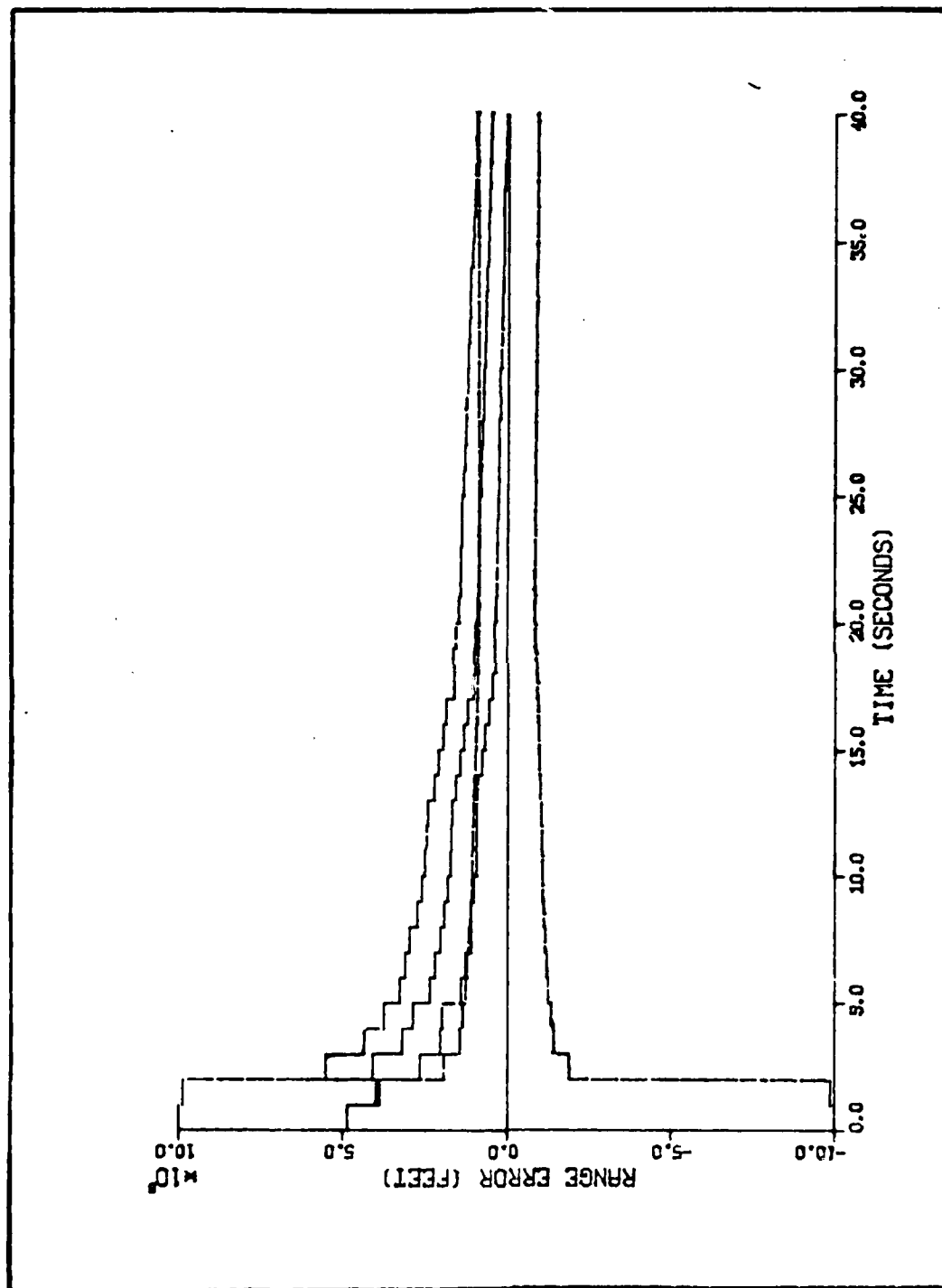


Figure K-1. Algorithm 5, Range Error Versus Time, No Pseudonoise in Range Channel

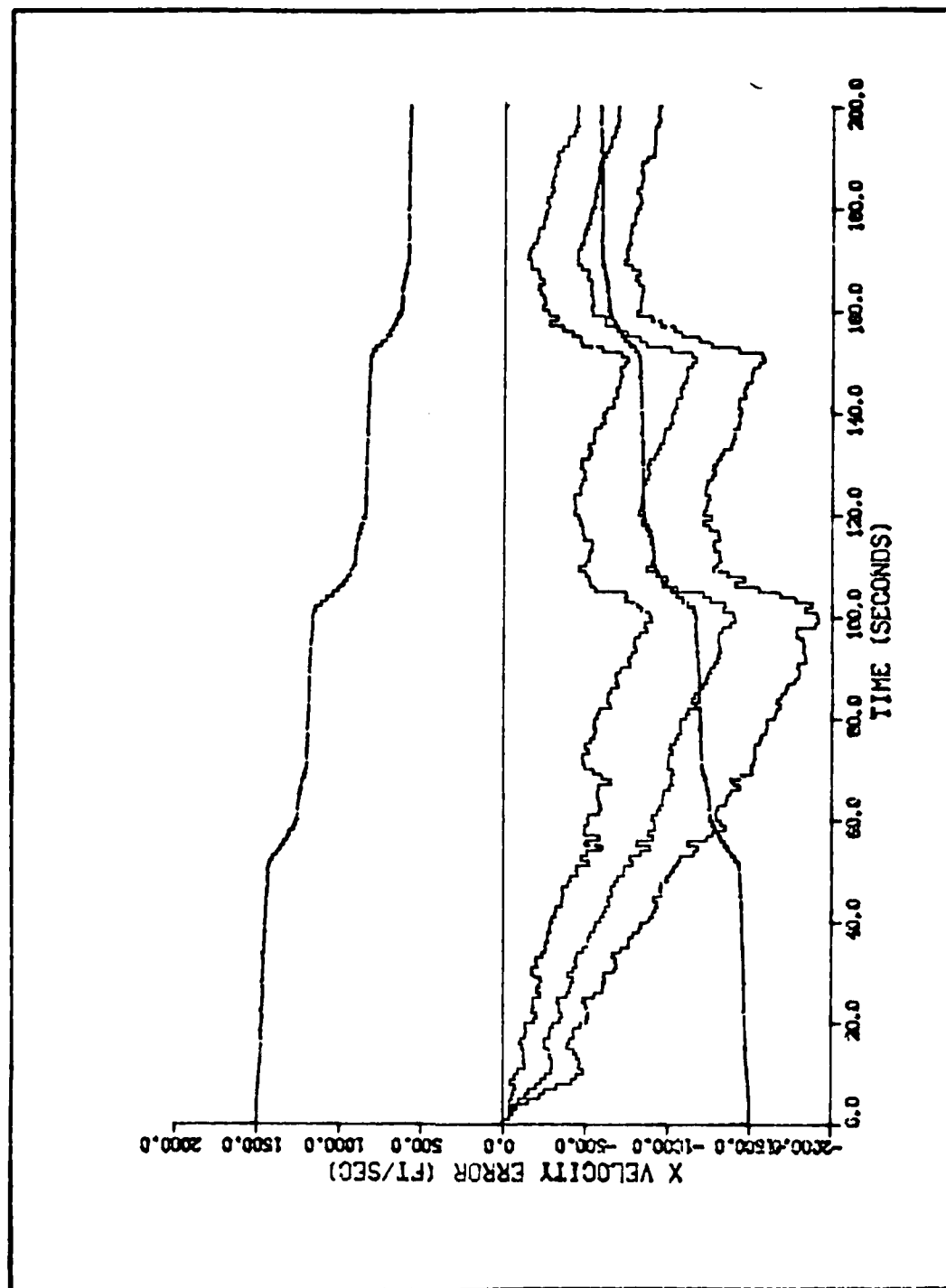


Figure K-2. Algorithm 5, Velocity Error in the X Direction Versus Time,
No Pseudonoise in the Range Channel

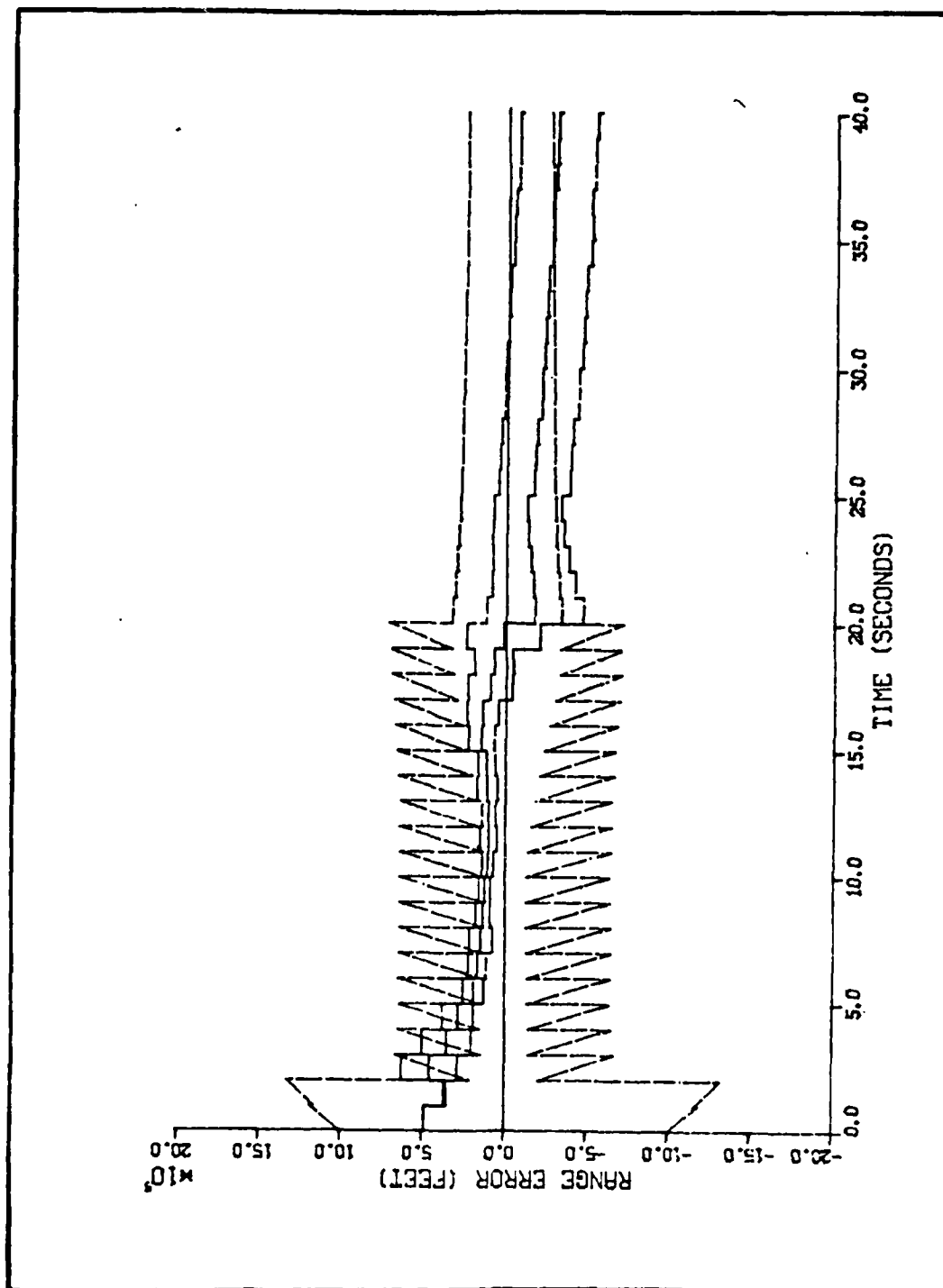


Figure K-3. Algorithm 5, Range Error Versus Time, Pseudonoise in the Range Channel for the First 20 Seconds

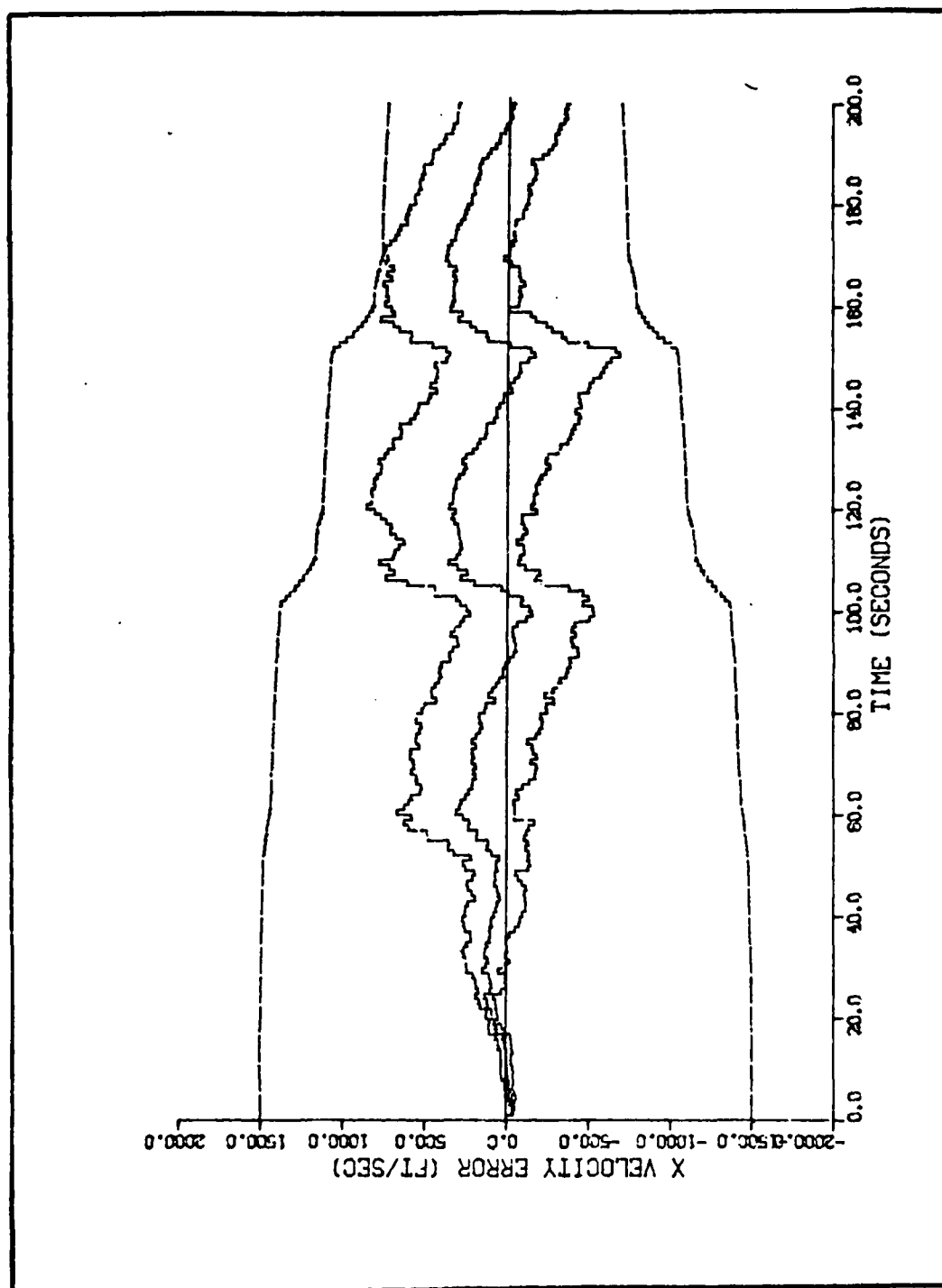


Figure K-4. Algorithm 5, Velocity Error in the X Direction Versus Time, Pseudonoise in the Range Channel for the First 20 seconds

APPENDIX L
GRAPHICAL RESULTS OF ALGORITHM FIVE

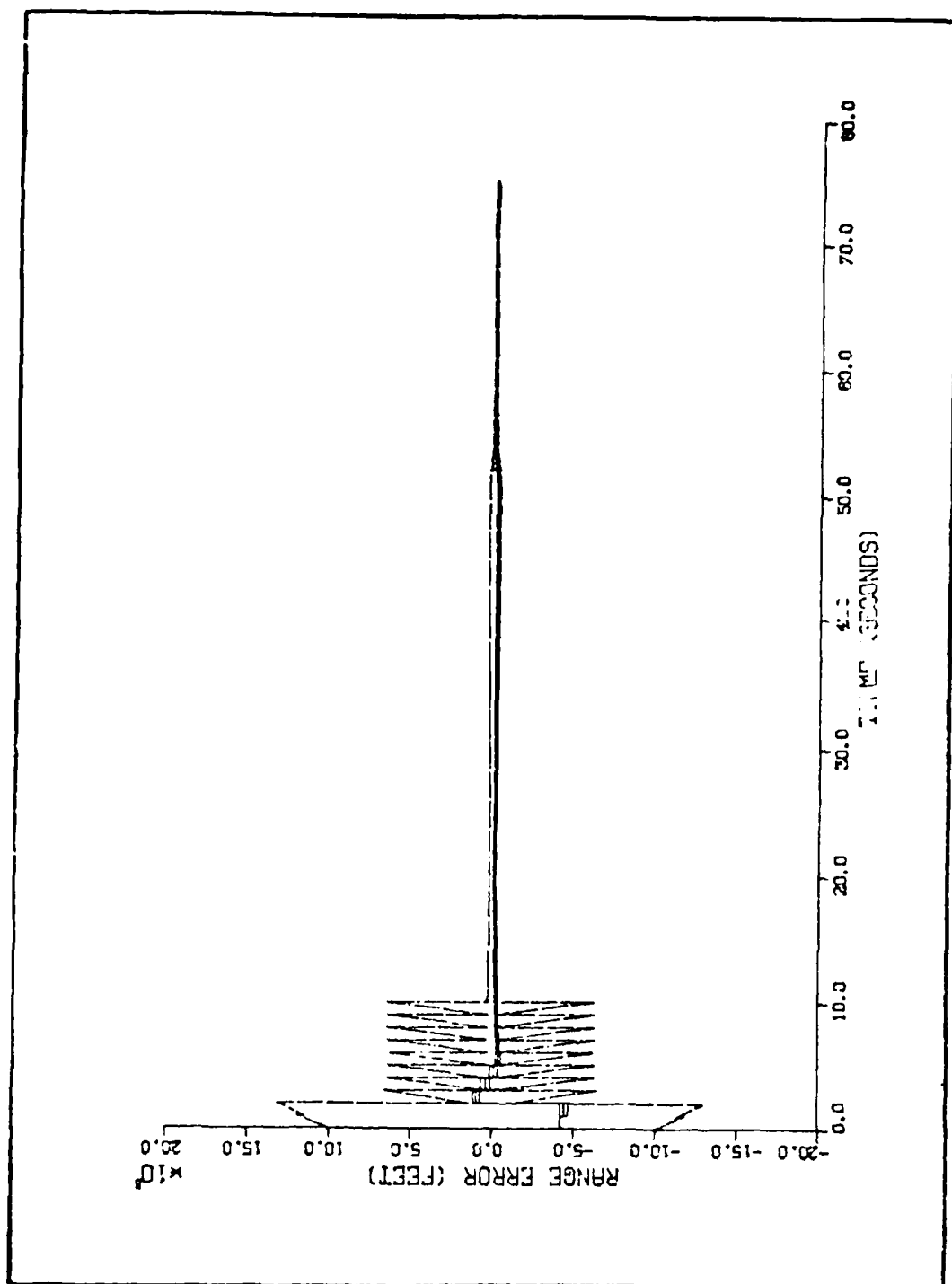


Figure I-1. Algorithm 5, Range Error Versus Time, Target Profile 1

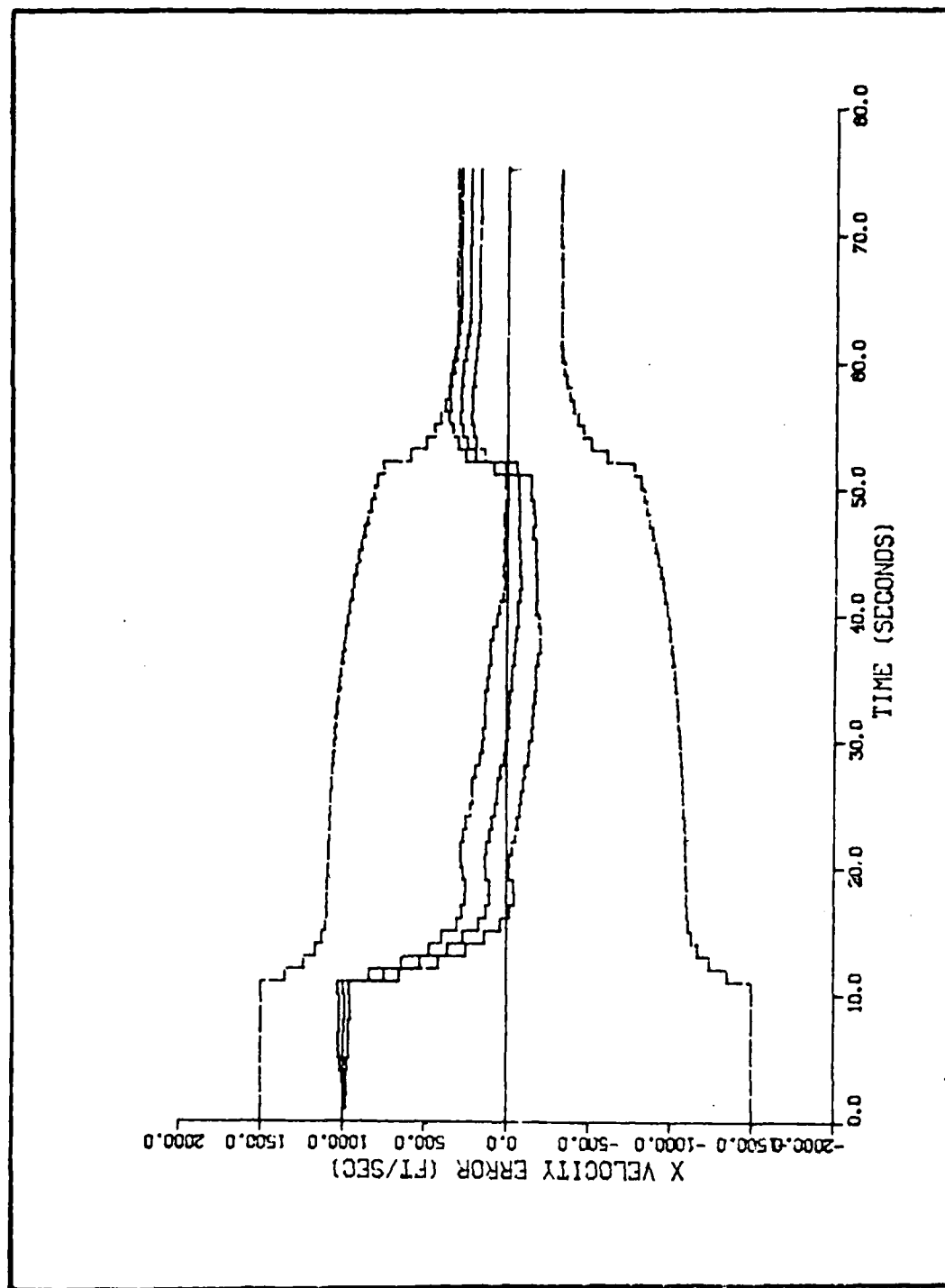


Figure L-2. Algorithm 5, Velocity Error in the X Direction Versus Time,
Target Profile 1

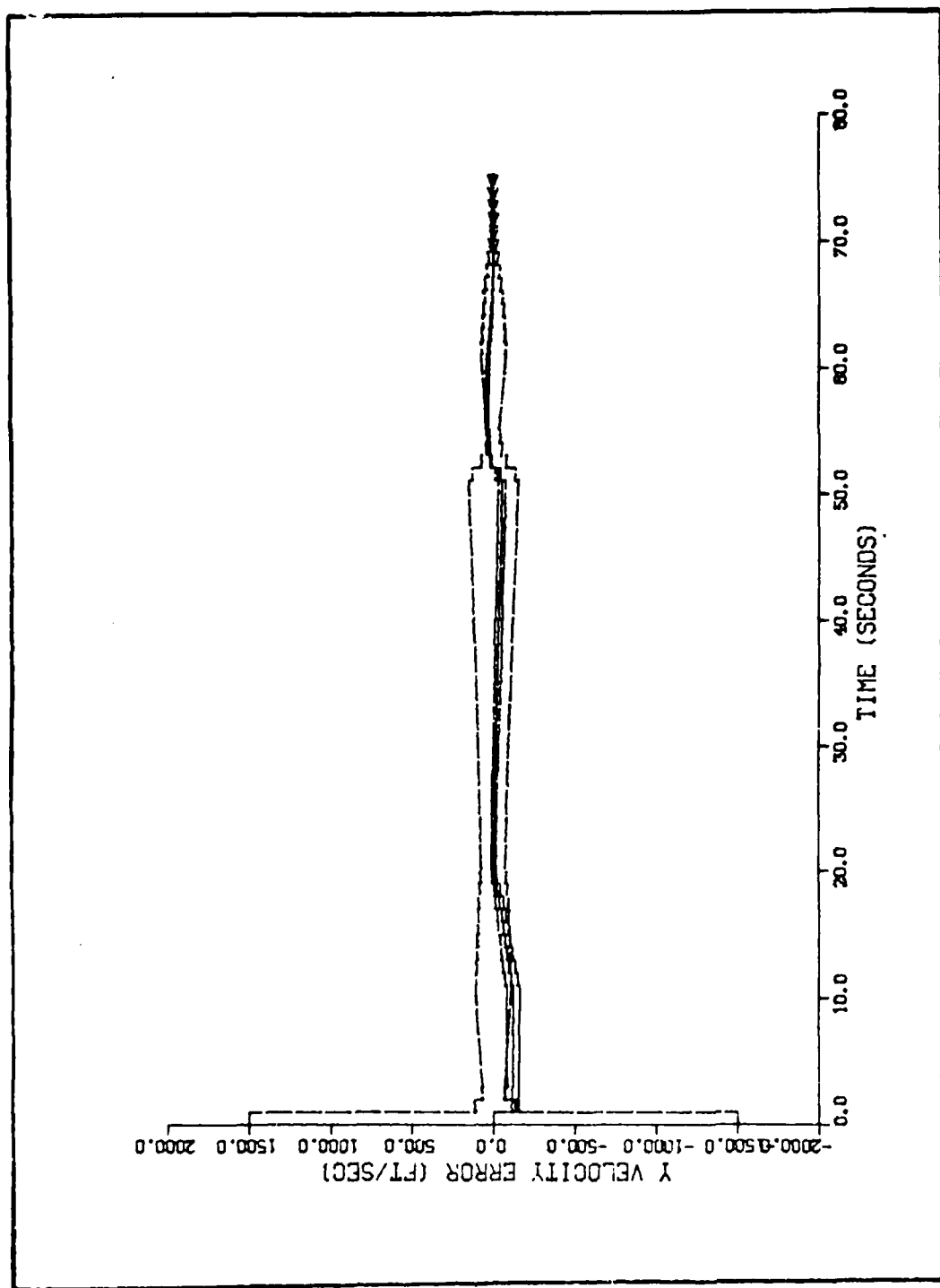


Figure L-3. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 1

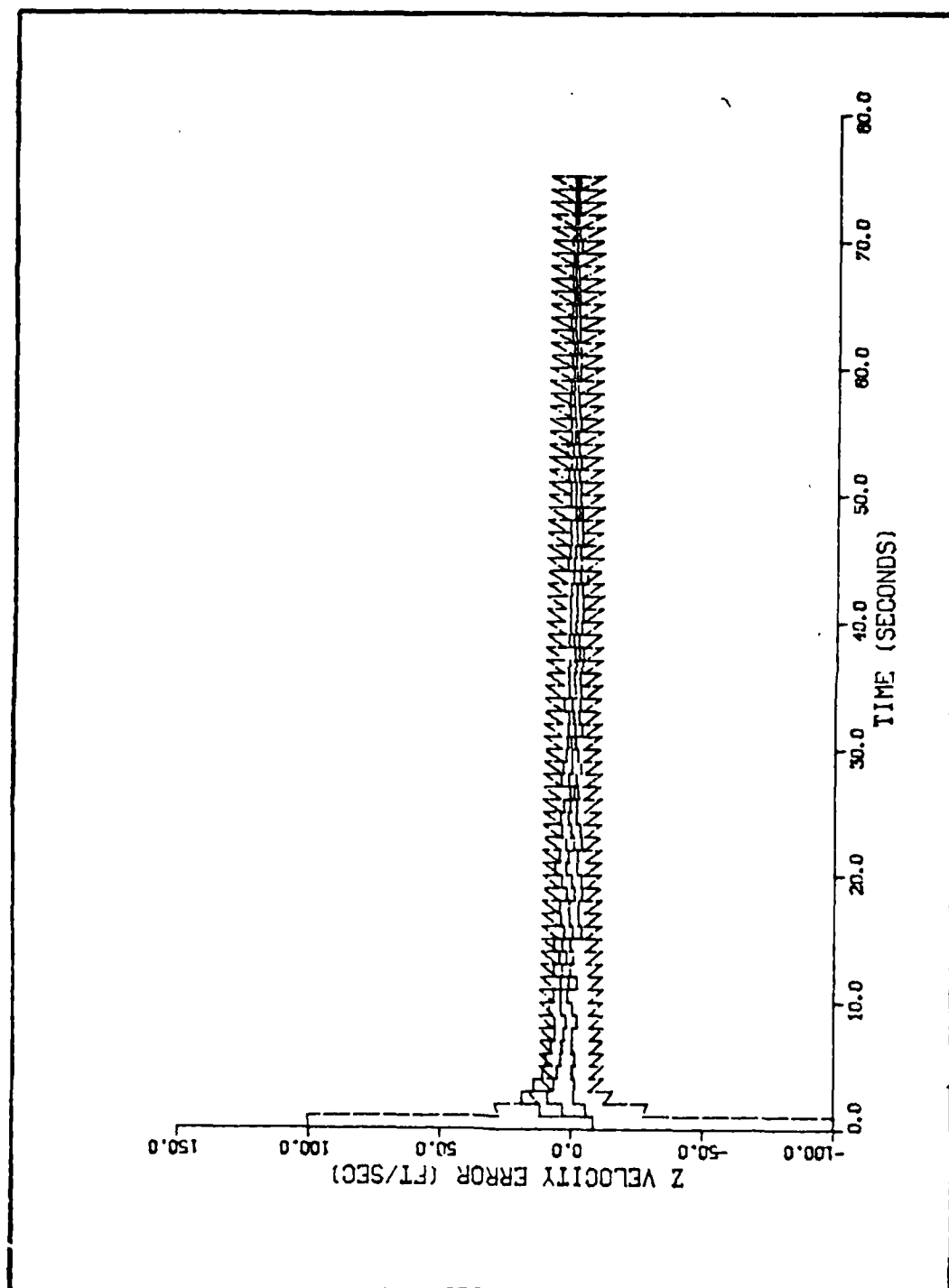


Figure L-3A. Algorithm 5, Velocity Error in the Z Direction Versus Time, Target Profile 1

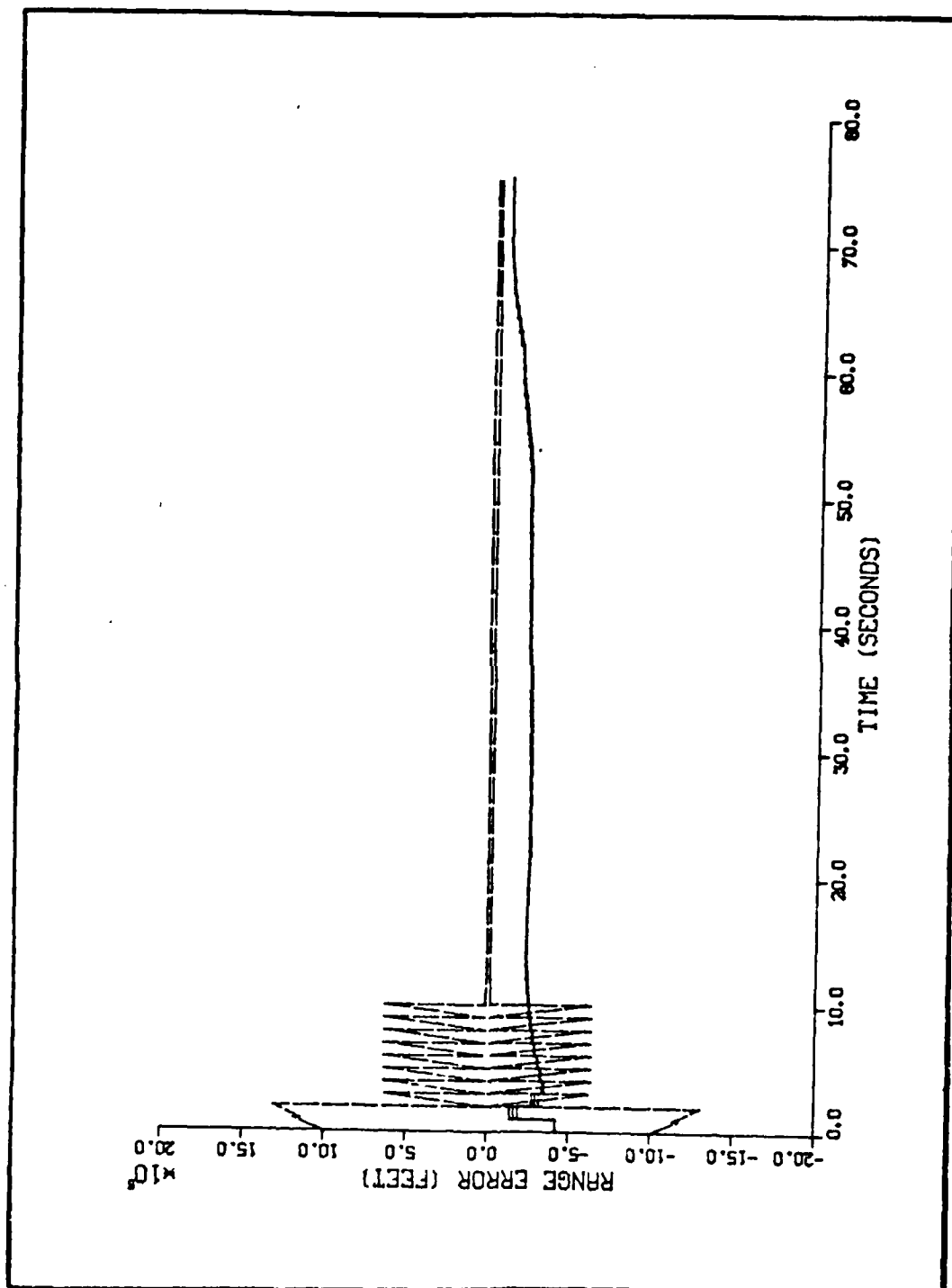


Figure L-4. Algorithm 5, Range Error Versus Time, Target Profile 2

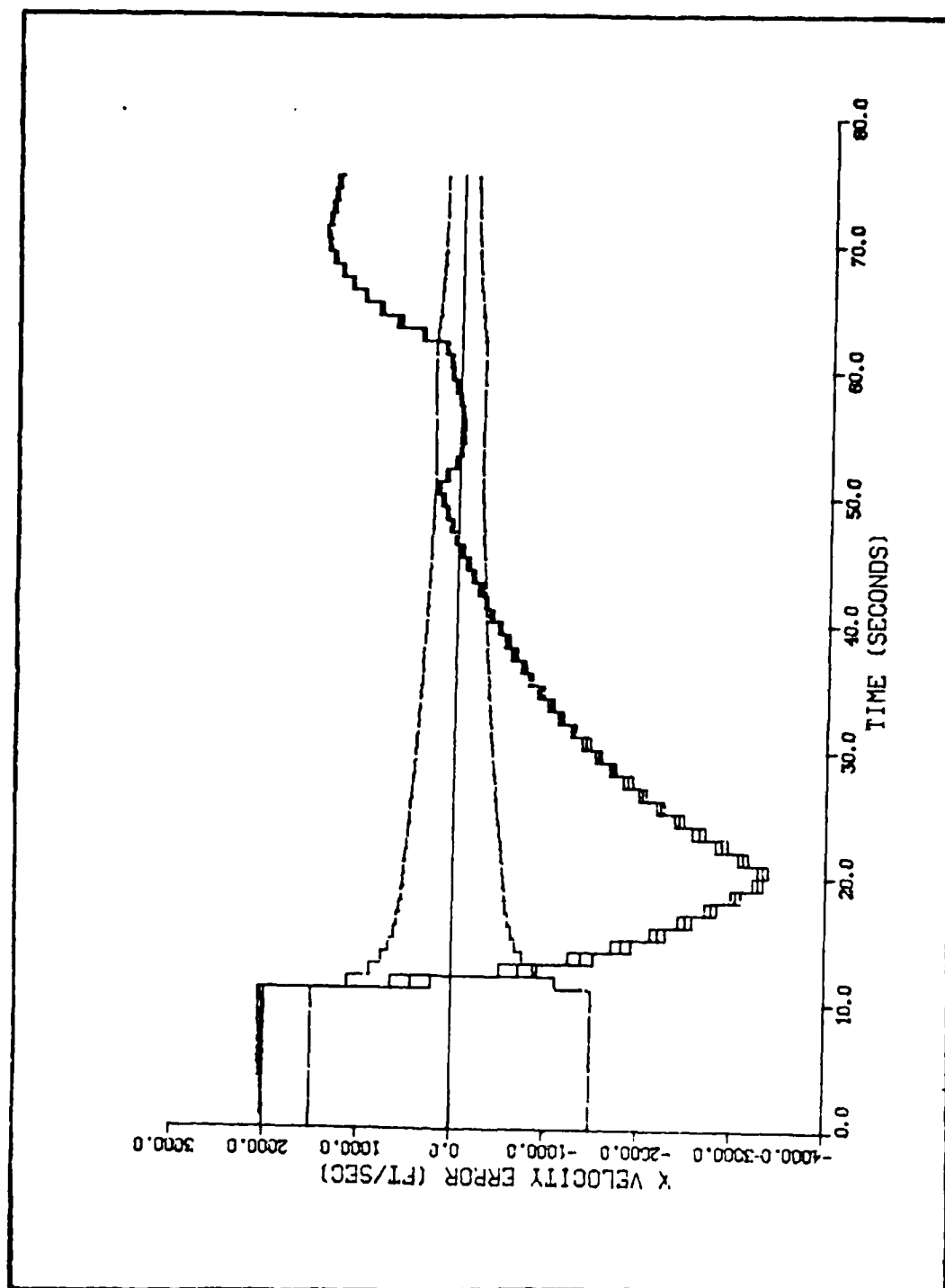


Figure L-5. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 2

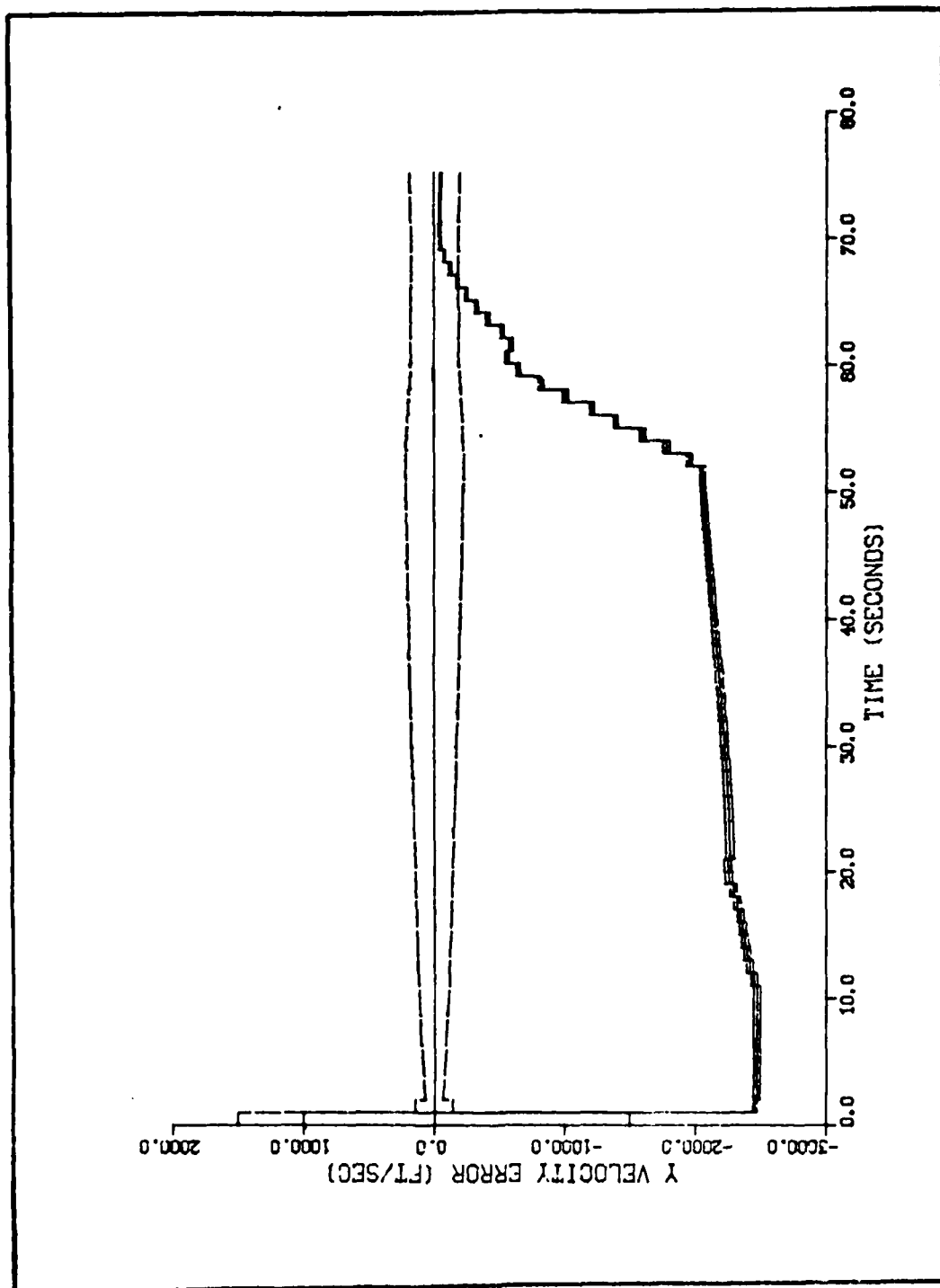


Figure L-6. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 2

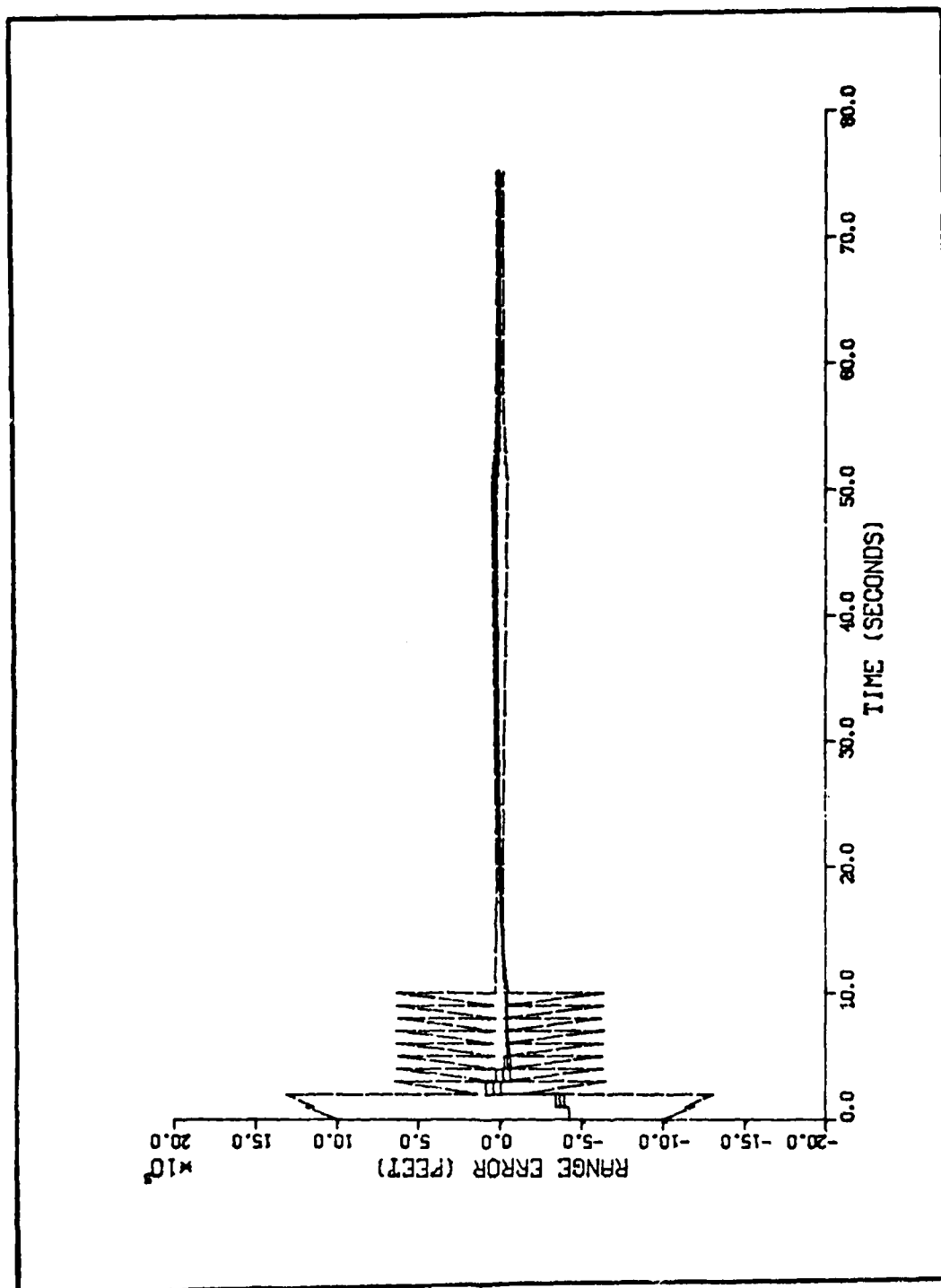


Figure L-7. Algorithm 5, Range Error Versus Time, Target Profile 3

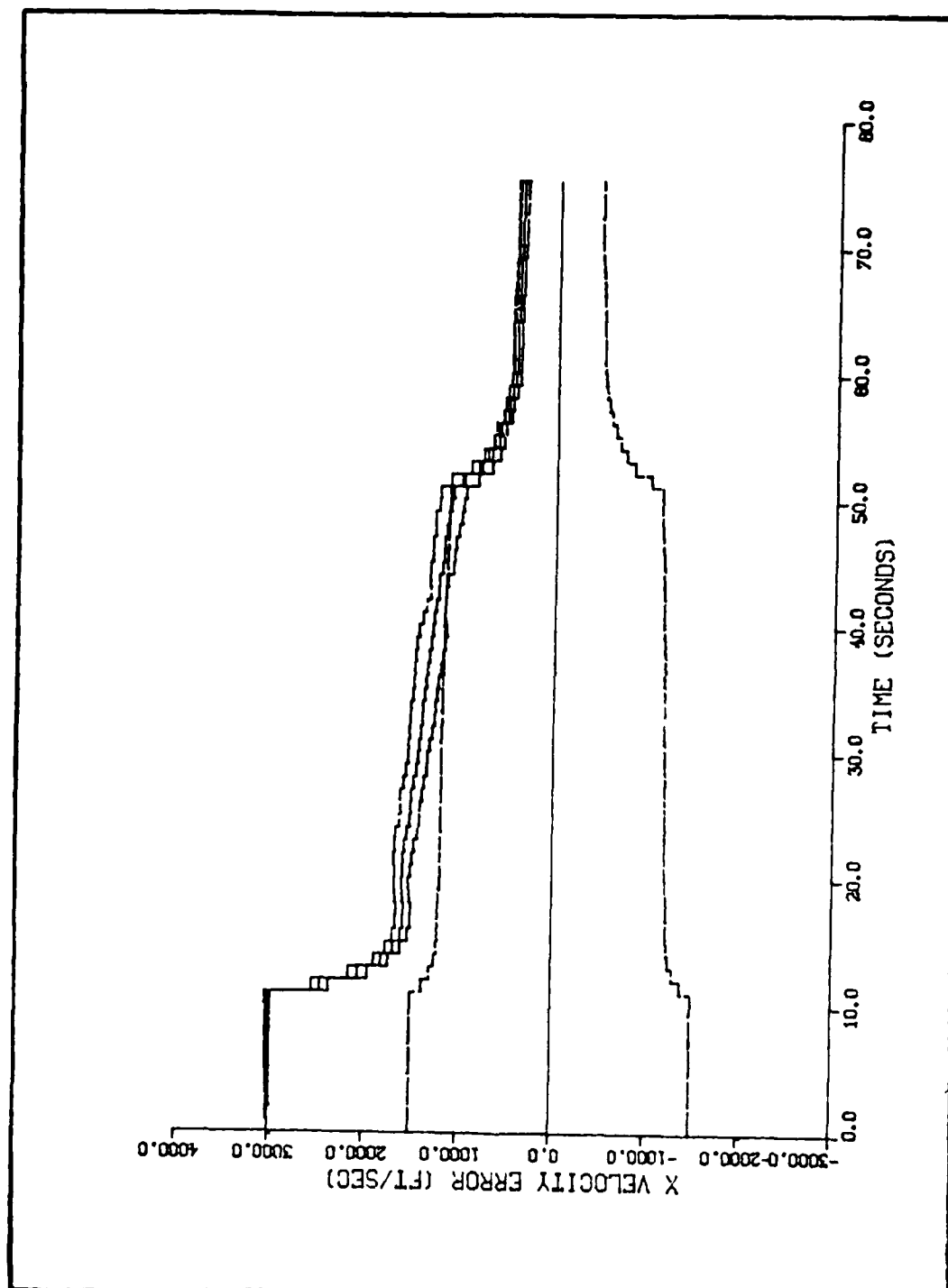


Figure L-8. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 3

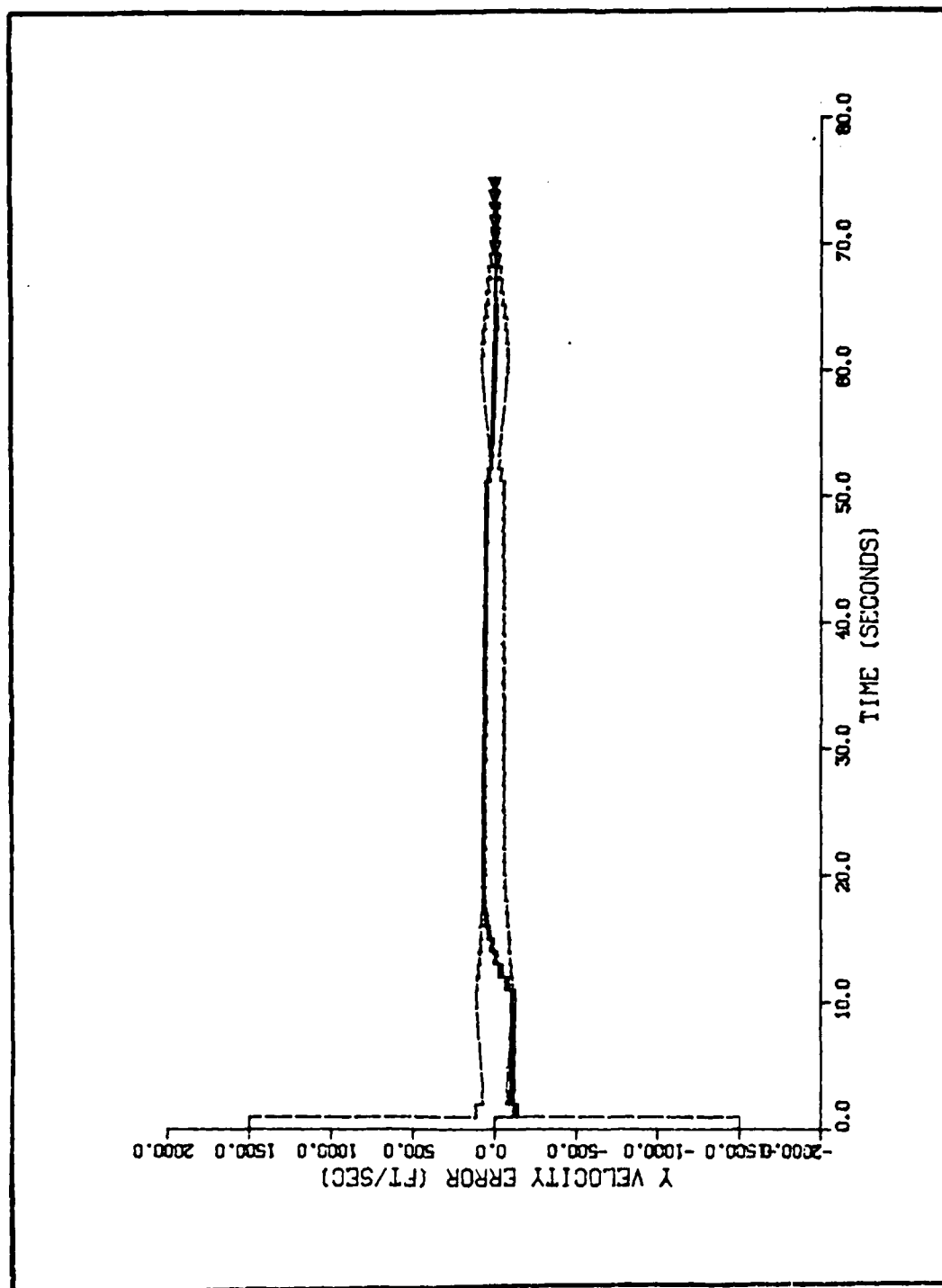


Figure L-9. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 3

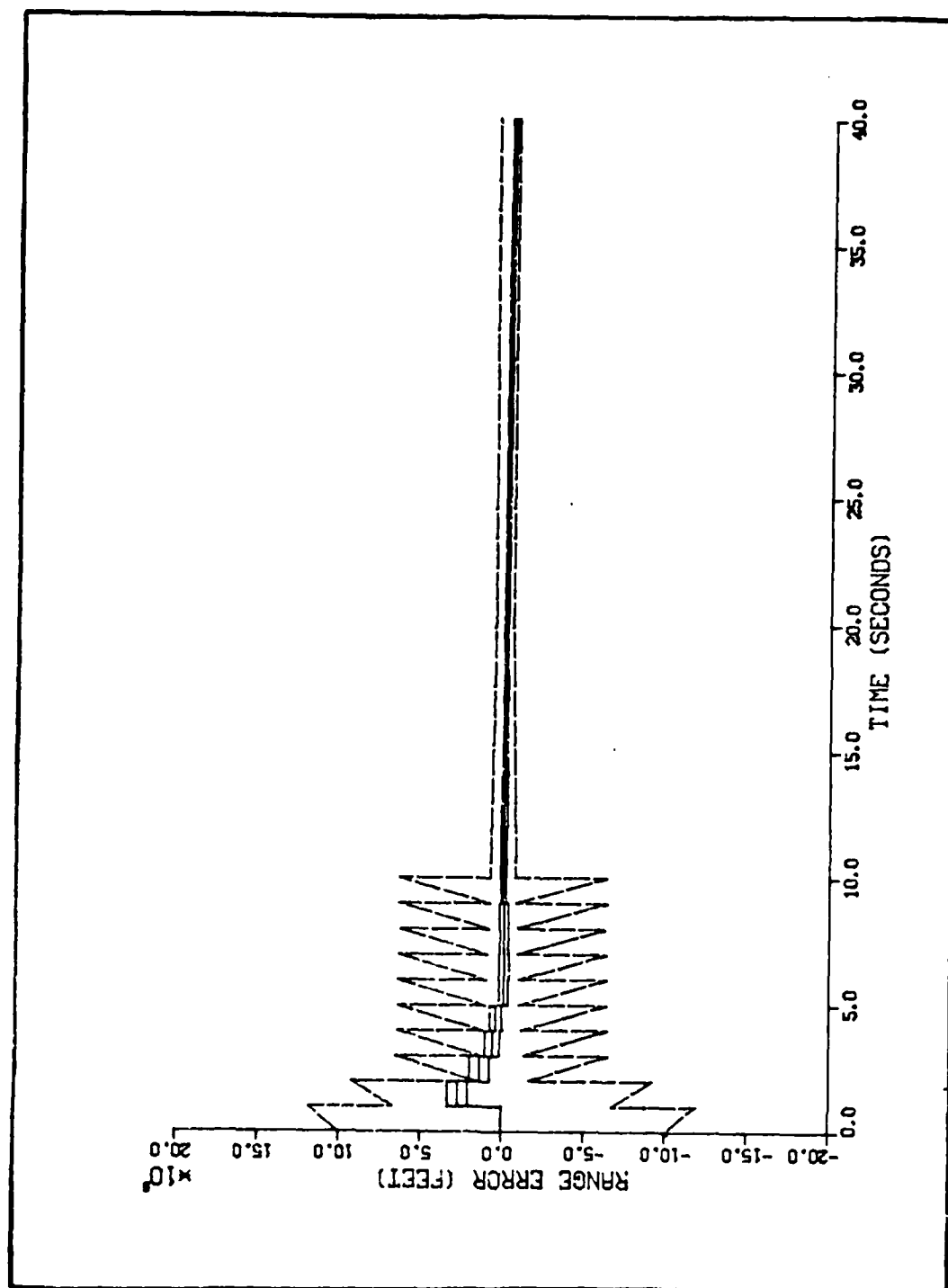


Figure L-10. Algorithm 5, Range Error Versus Time, Target Profile 4

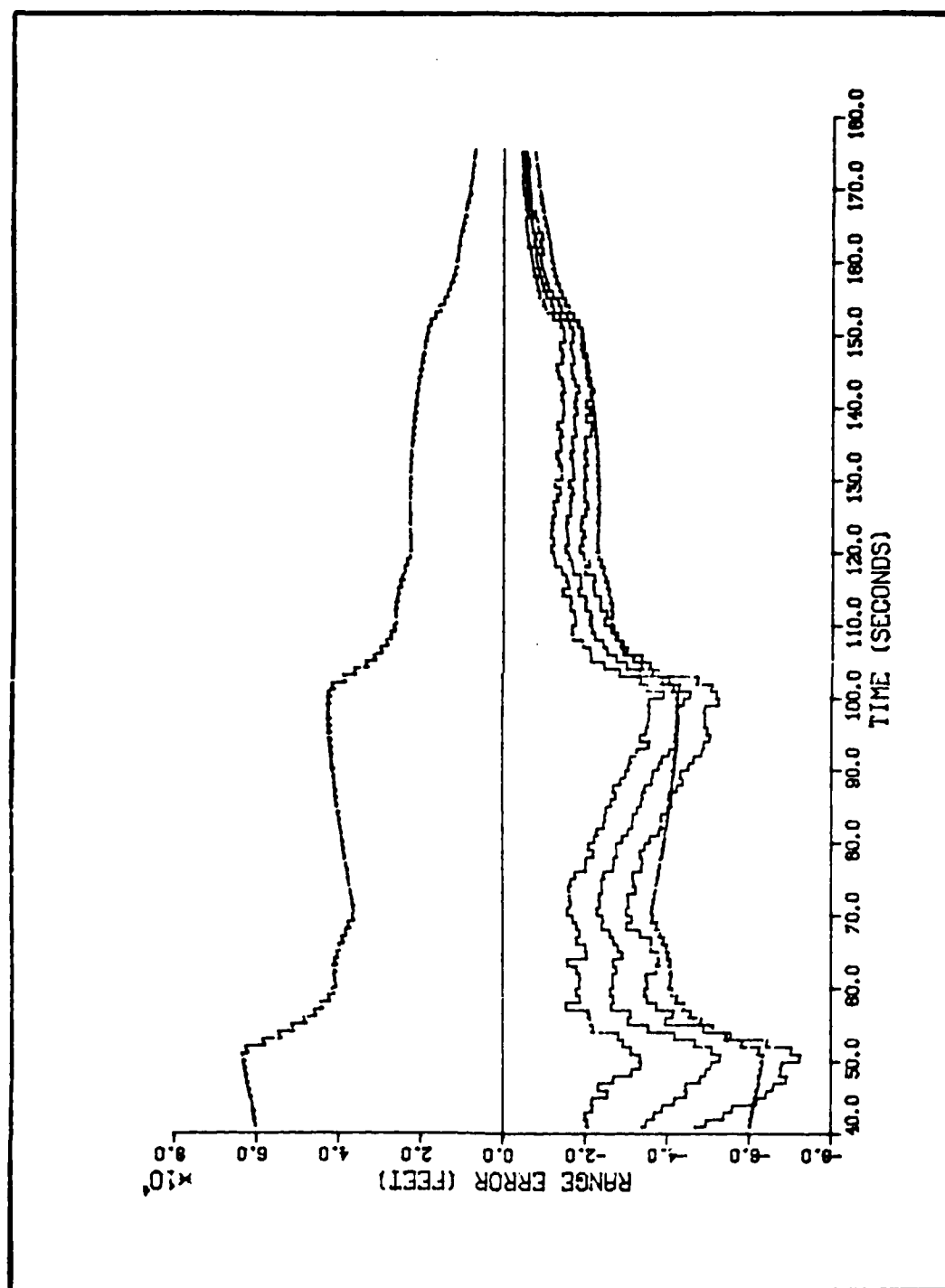


Figure L-11. Algorithm 5, Range Error Versus Time, Target Profile 4

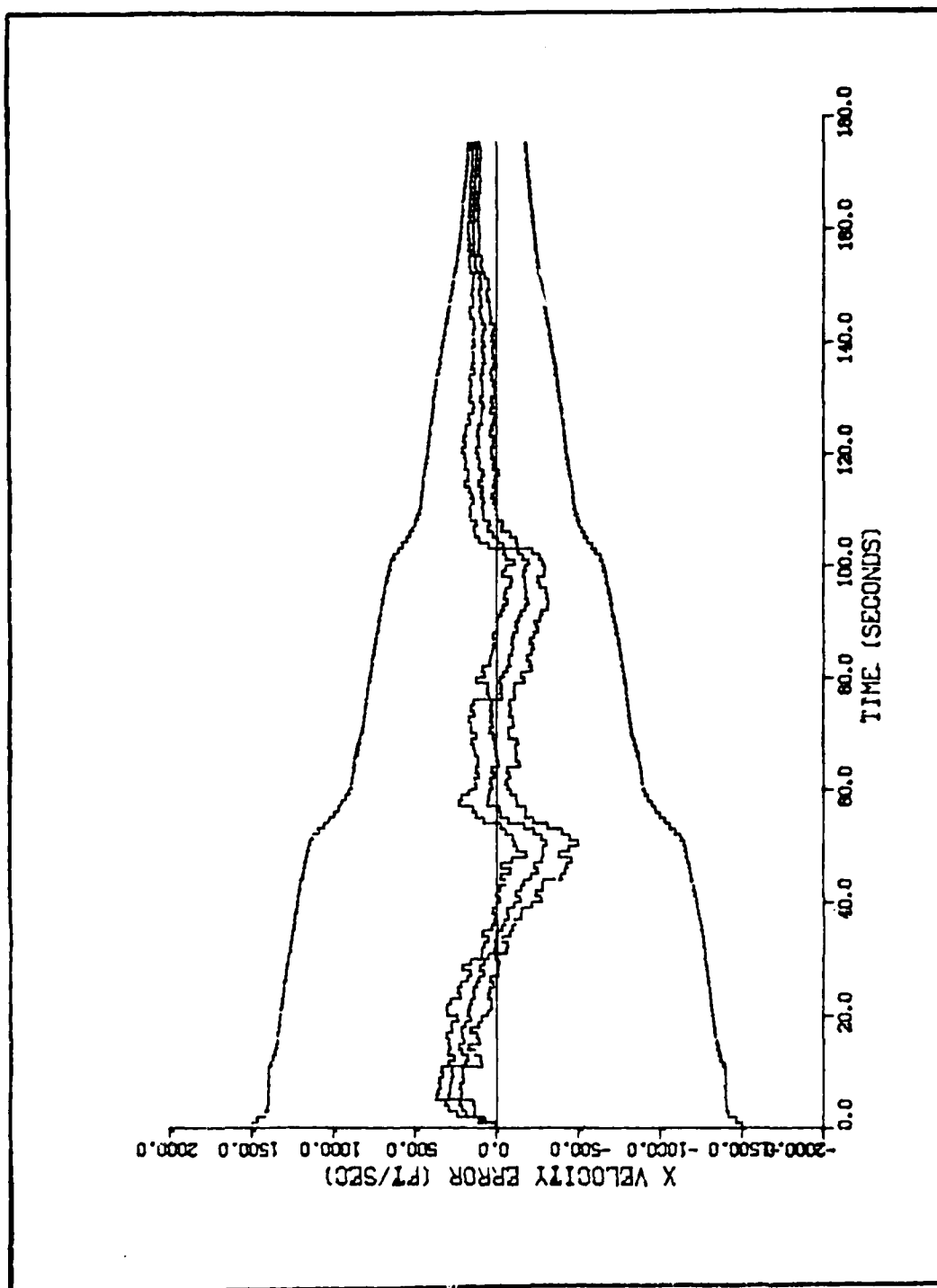


Figure 1-12. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 4

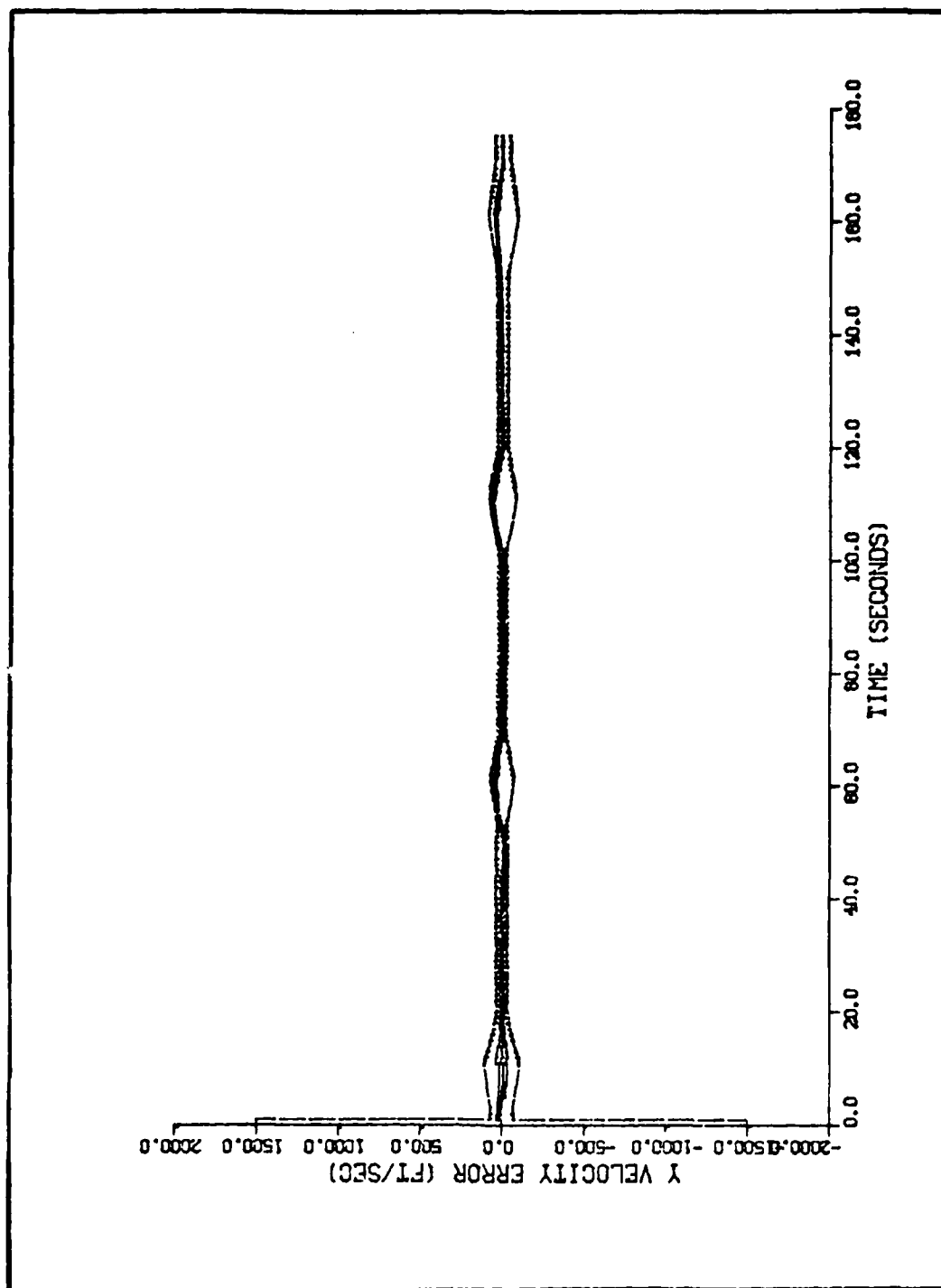


Figure L-13. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 4

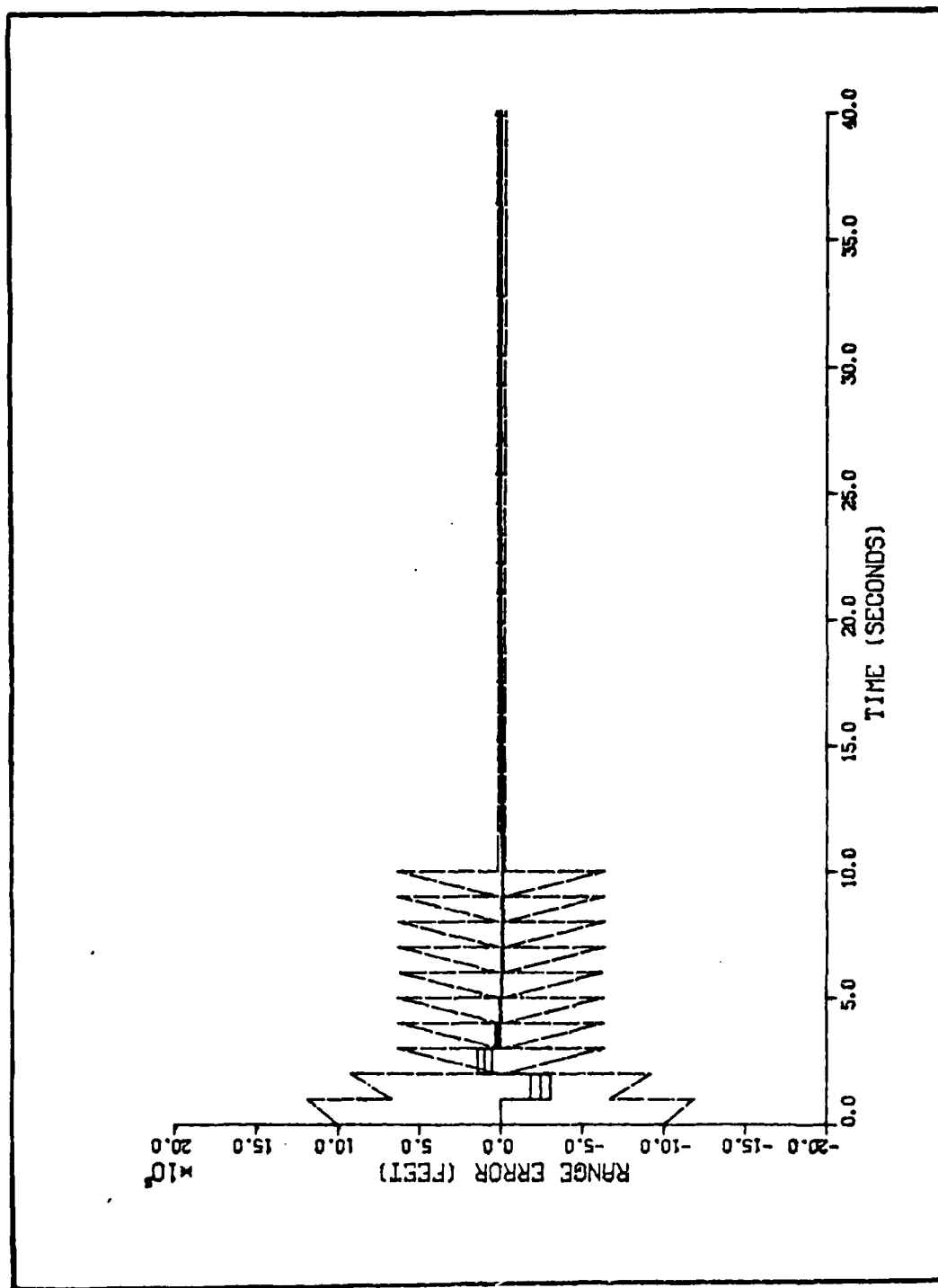


Figure L-14. Algorithm 5, Range Error Versus Time, Target Profile 5

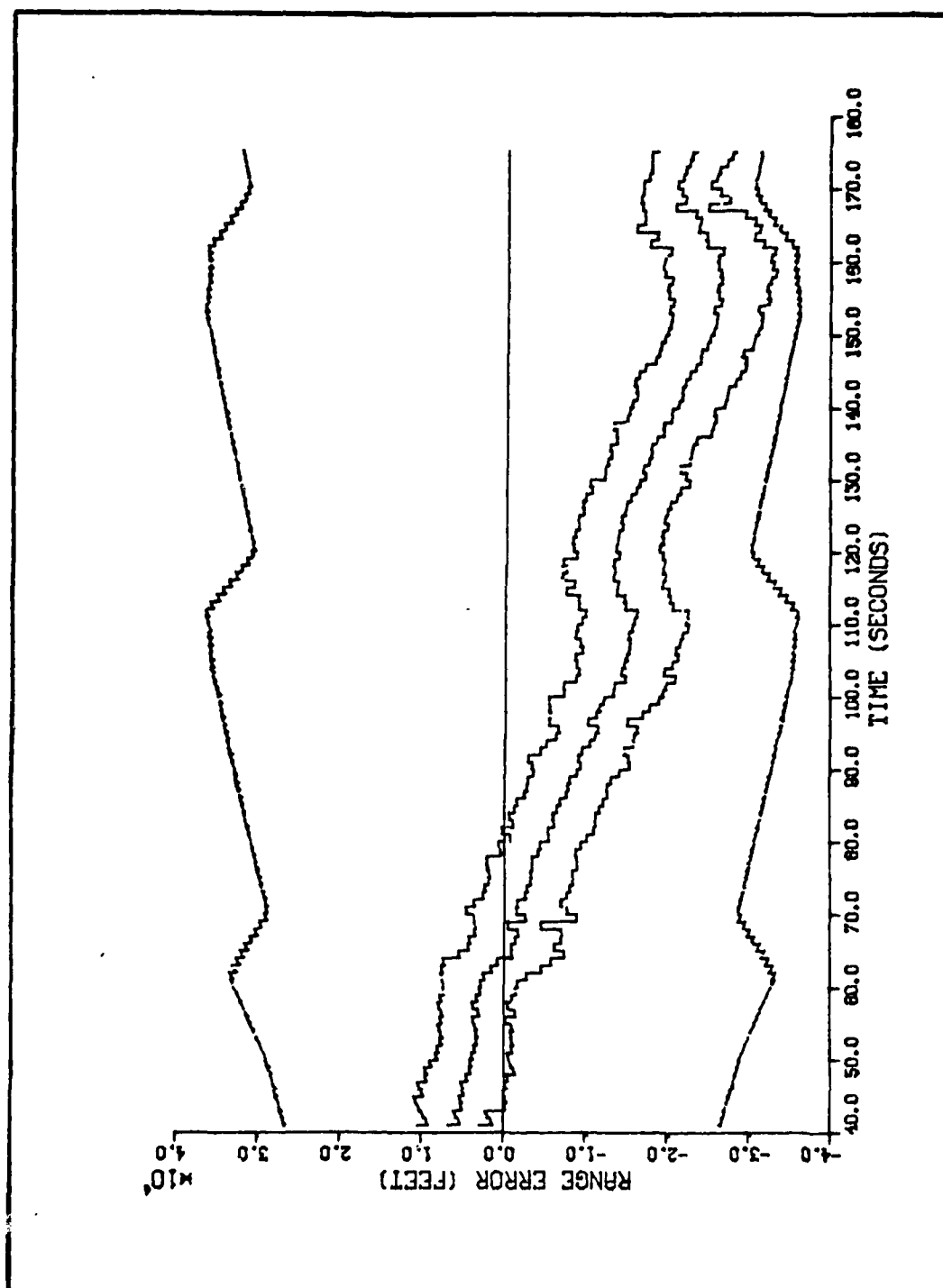


Figure L-15. Algorithm 5, Range Error Versus Time, Target Profile 5

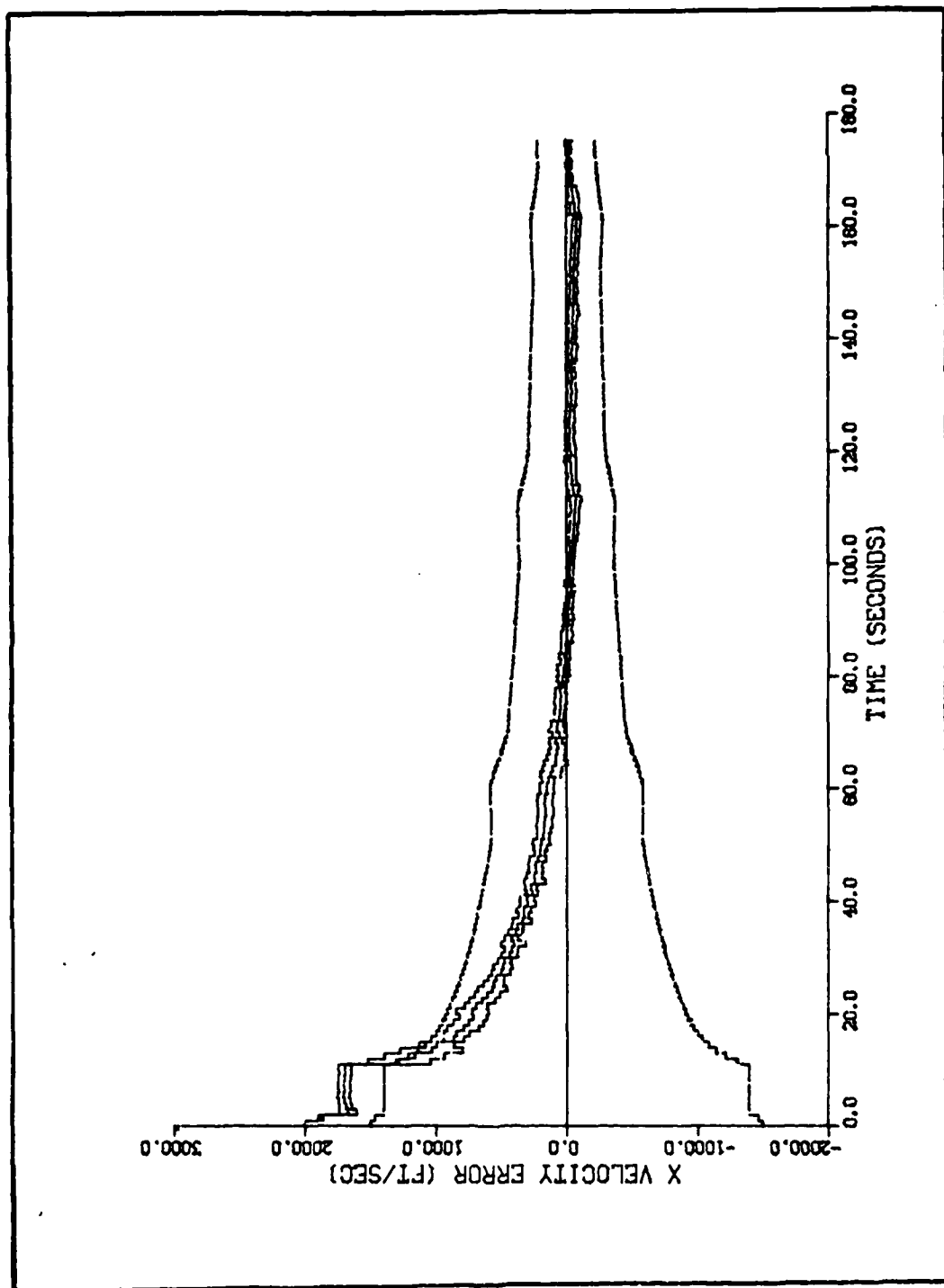


Figure L-16. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 5

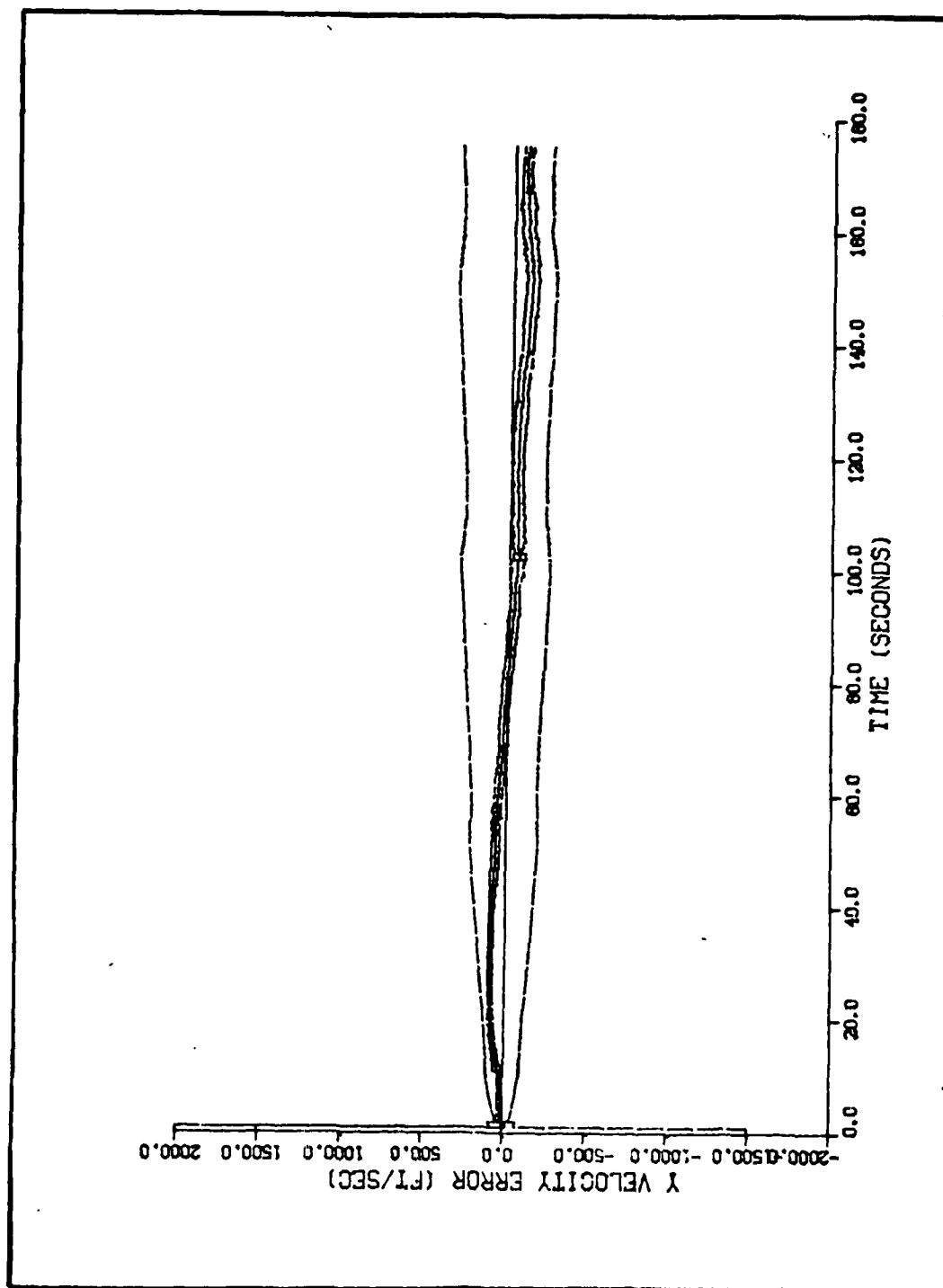


Figure L-17. Algorithm 5, Velocity in the Y Direction Versus Time, Target Profile 5

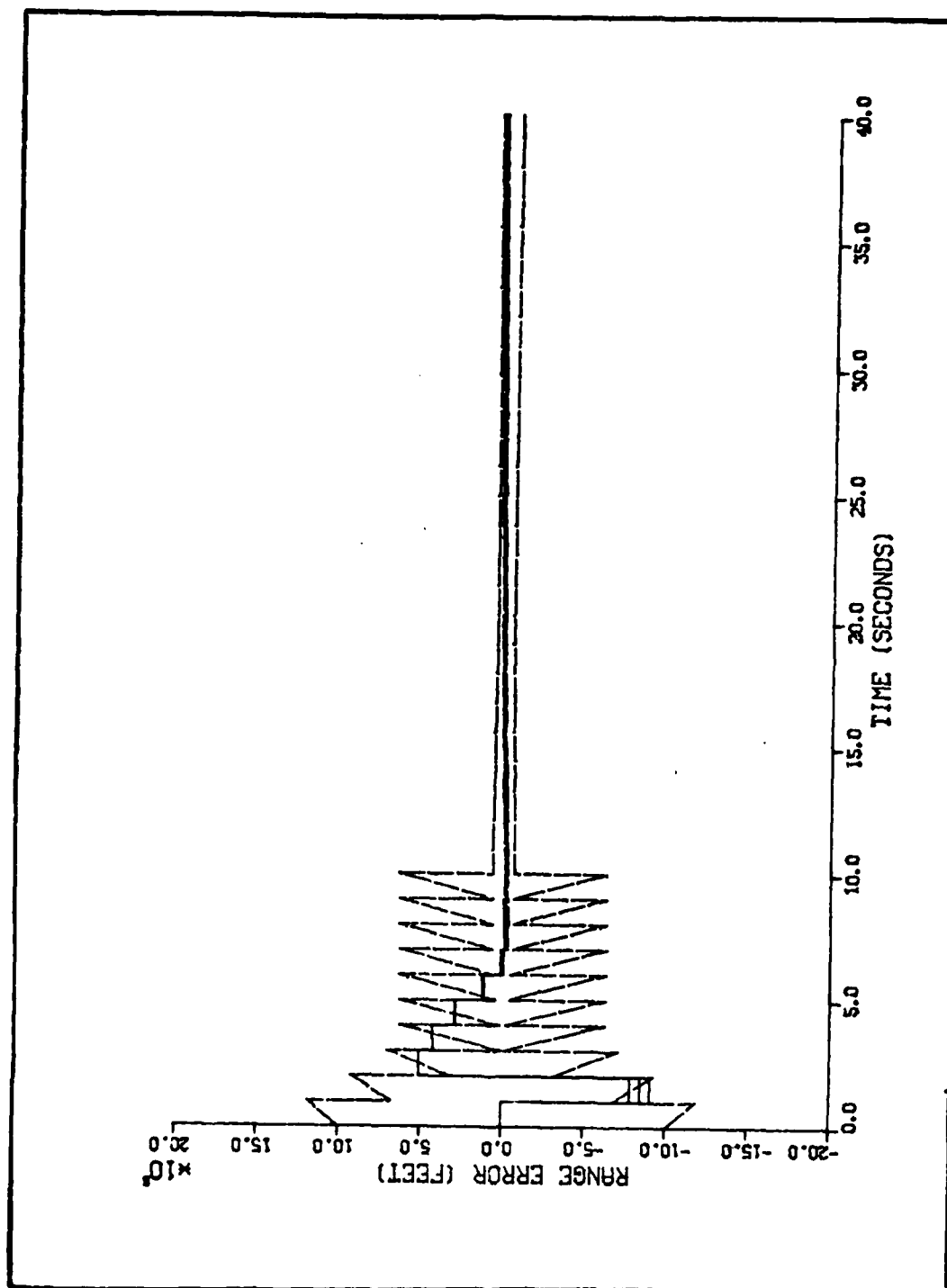


Figure L-18. Algorithm 5, Range Error Versus Time, Target Profile 6

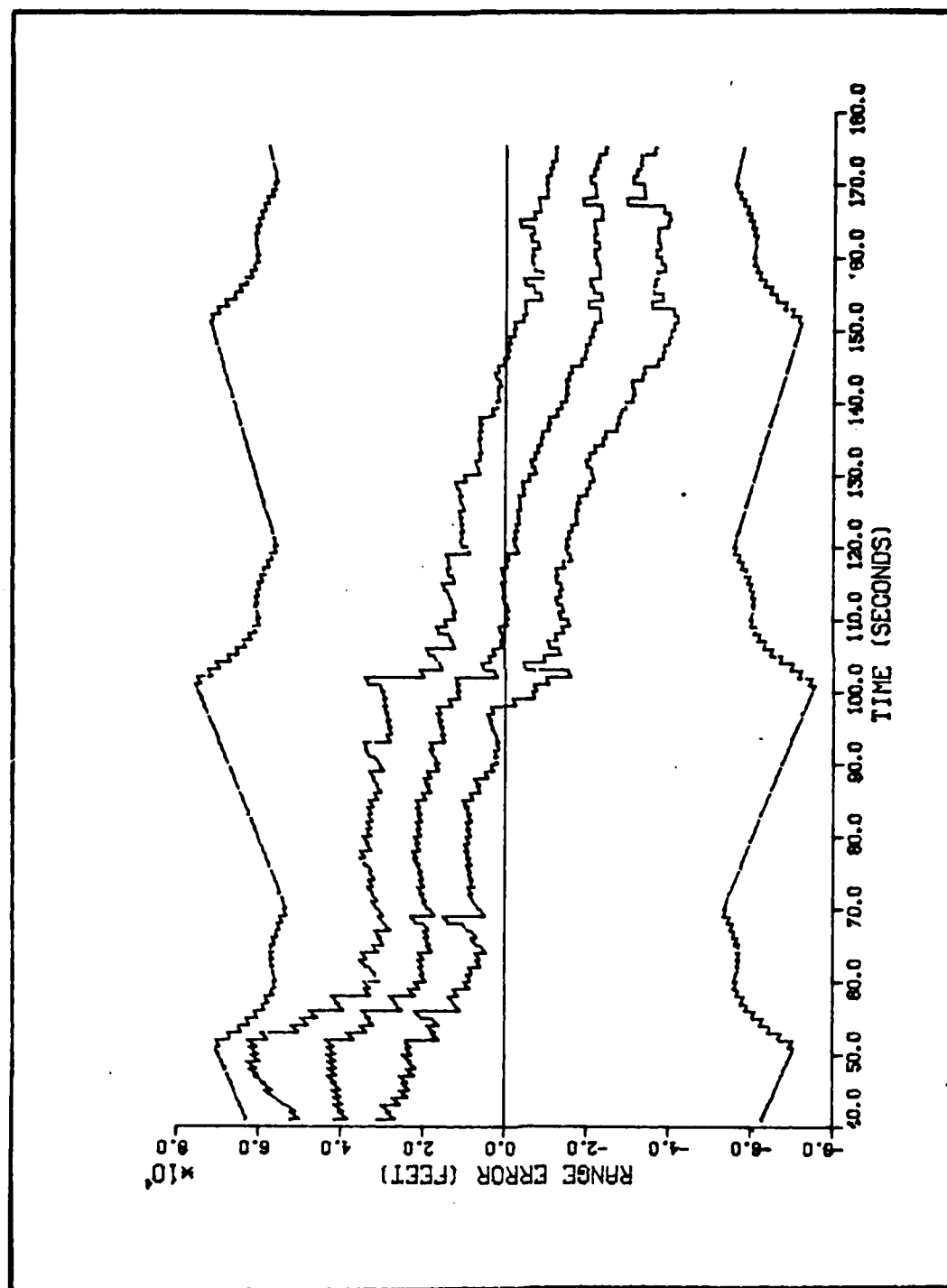


Figure L-19. Algorithm 5, Range Error Versus Time, Target Profile 6

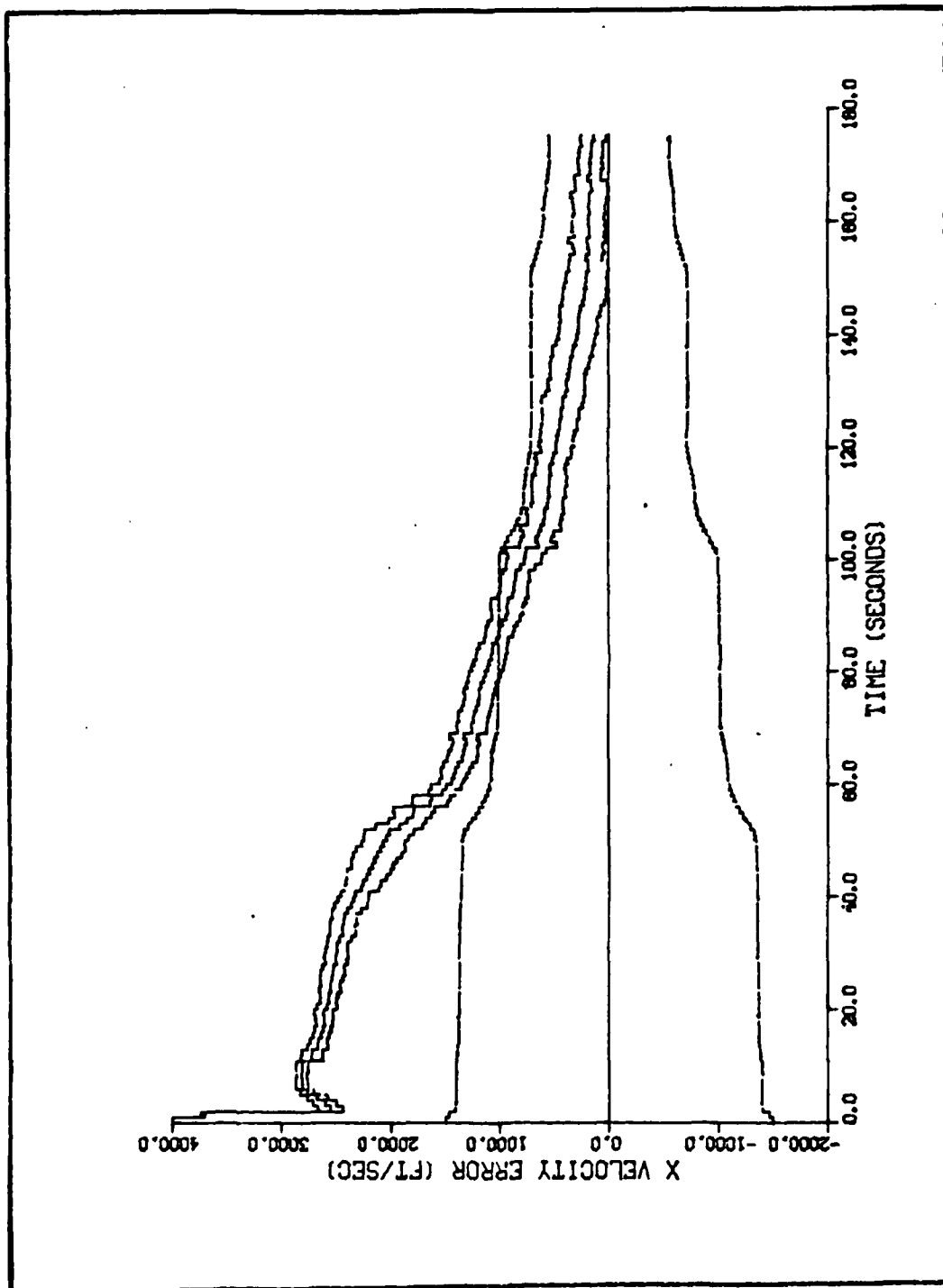


Figure L-20. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 6

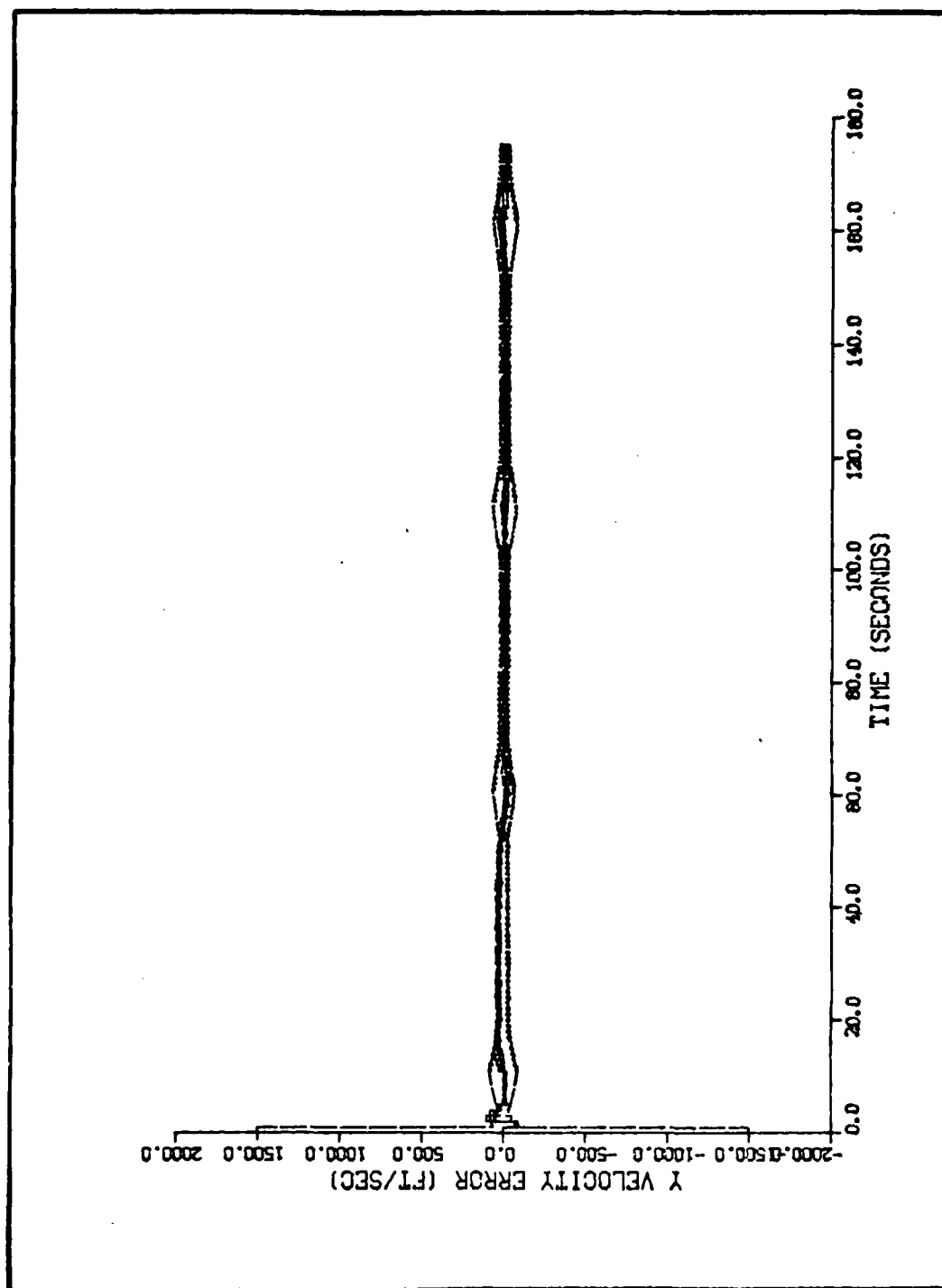


Figure L-21. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 6

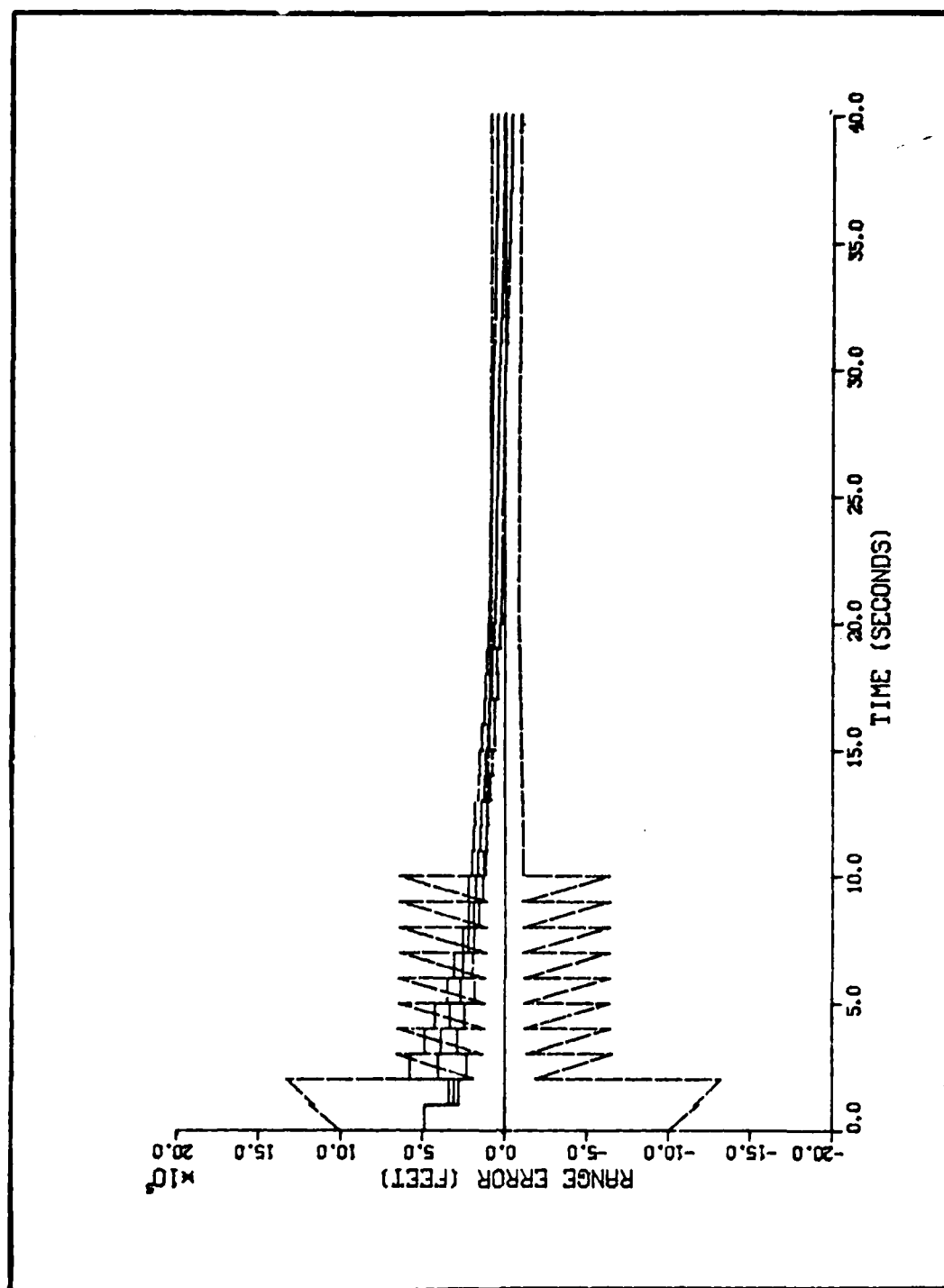


Figure L-22. Algorithm 5, Range Error Versus Time, Target Profile 7

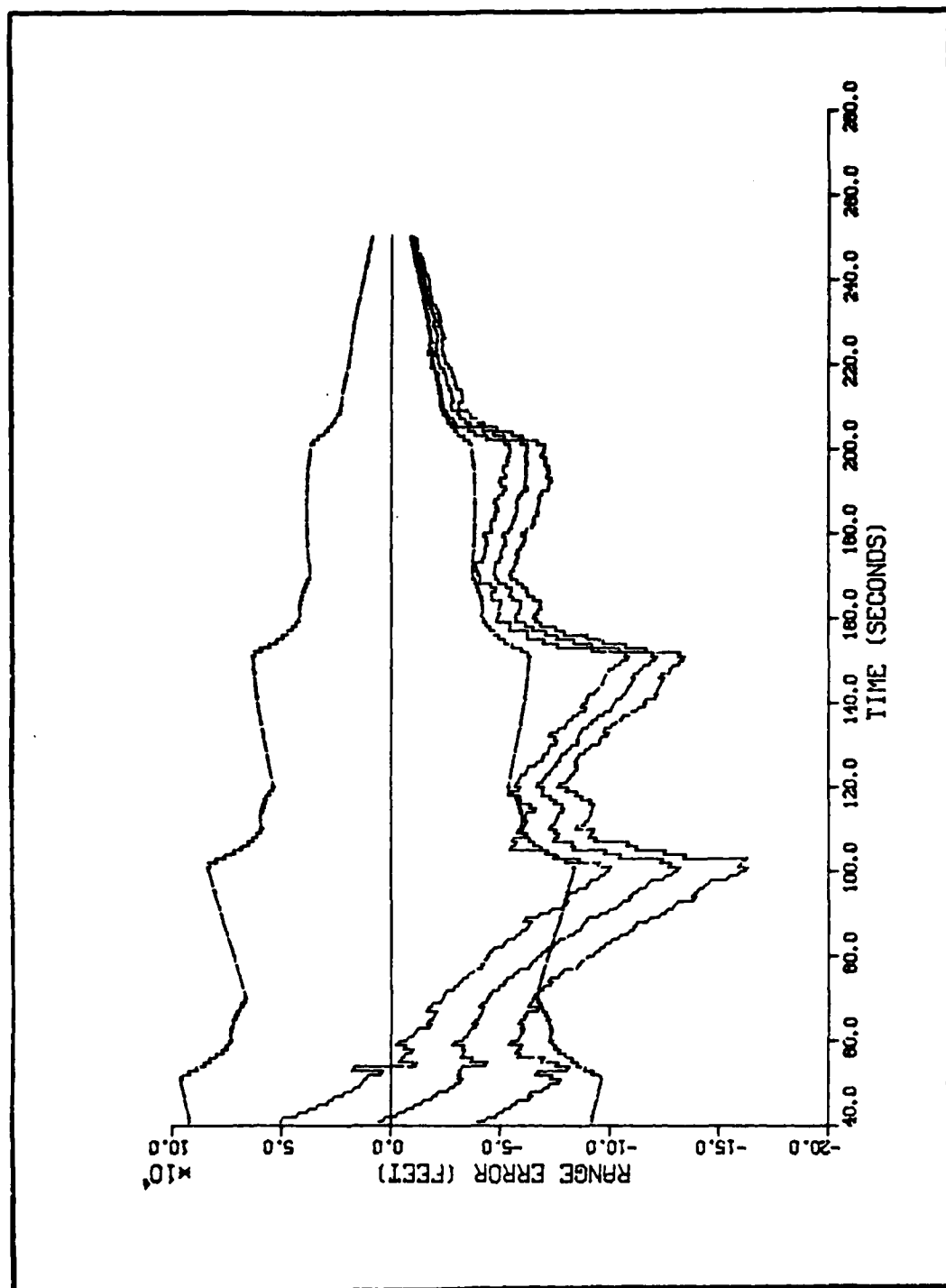


Figure L-23. Algorithm 5, Range Error Versus Time, Target Profile 7

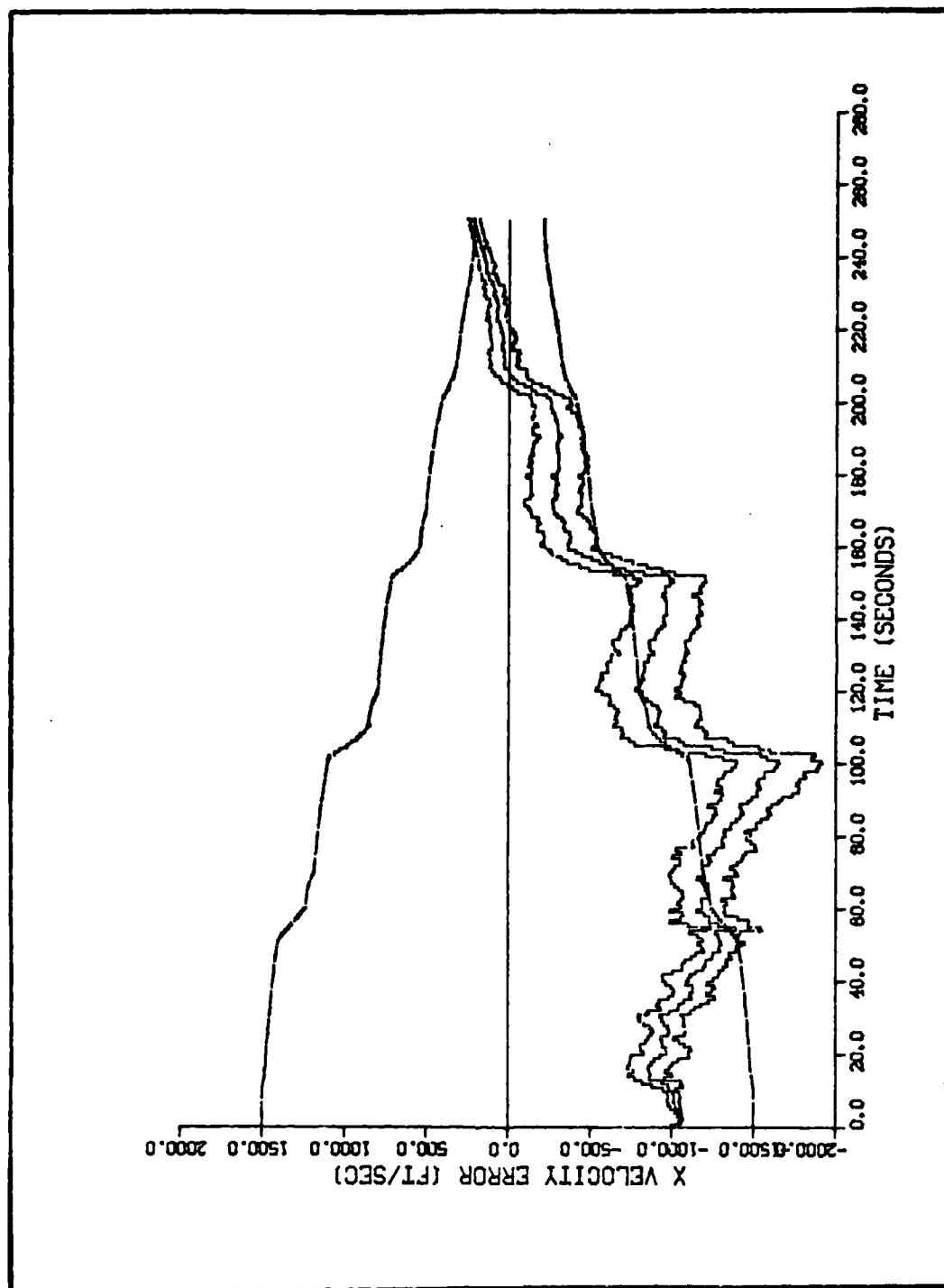


Figure L-24. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 7

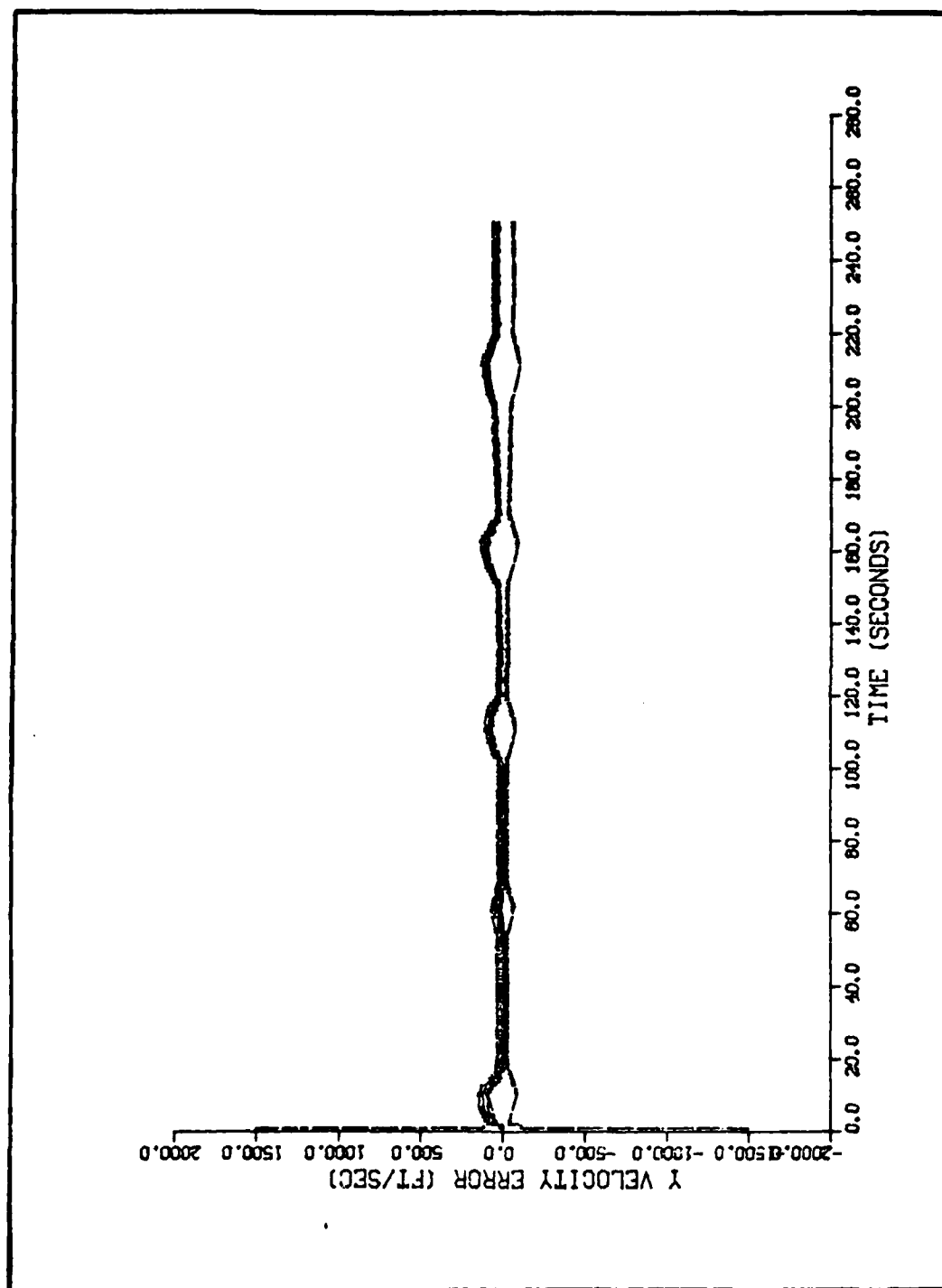


Figure L-25. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 7

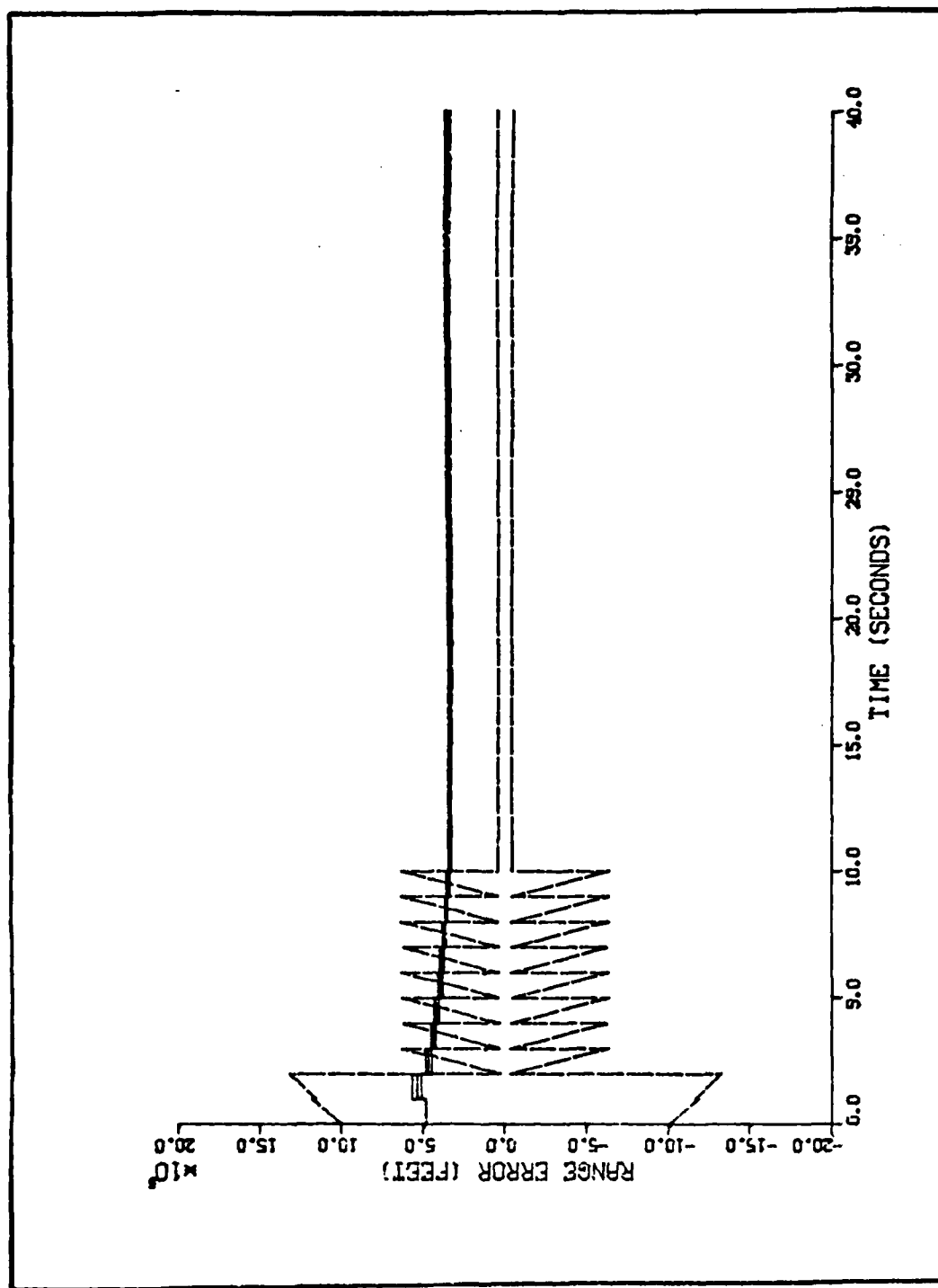


Figure L-26. Algorithm 5, Range Error Versus Time, Target Profile 8

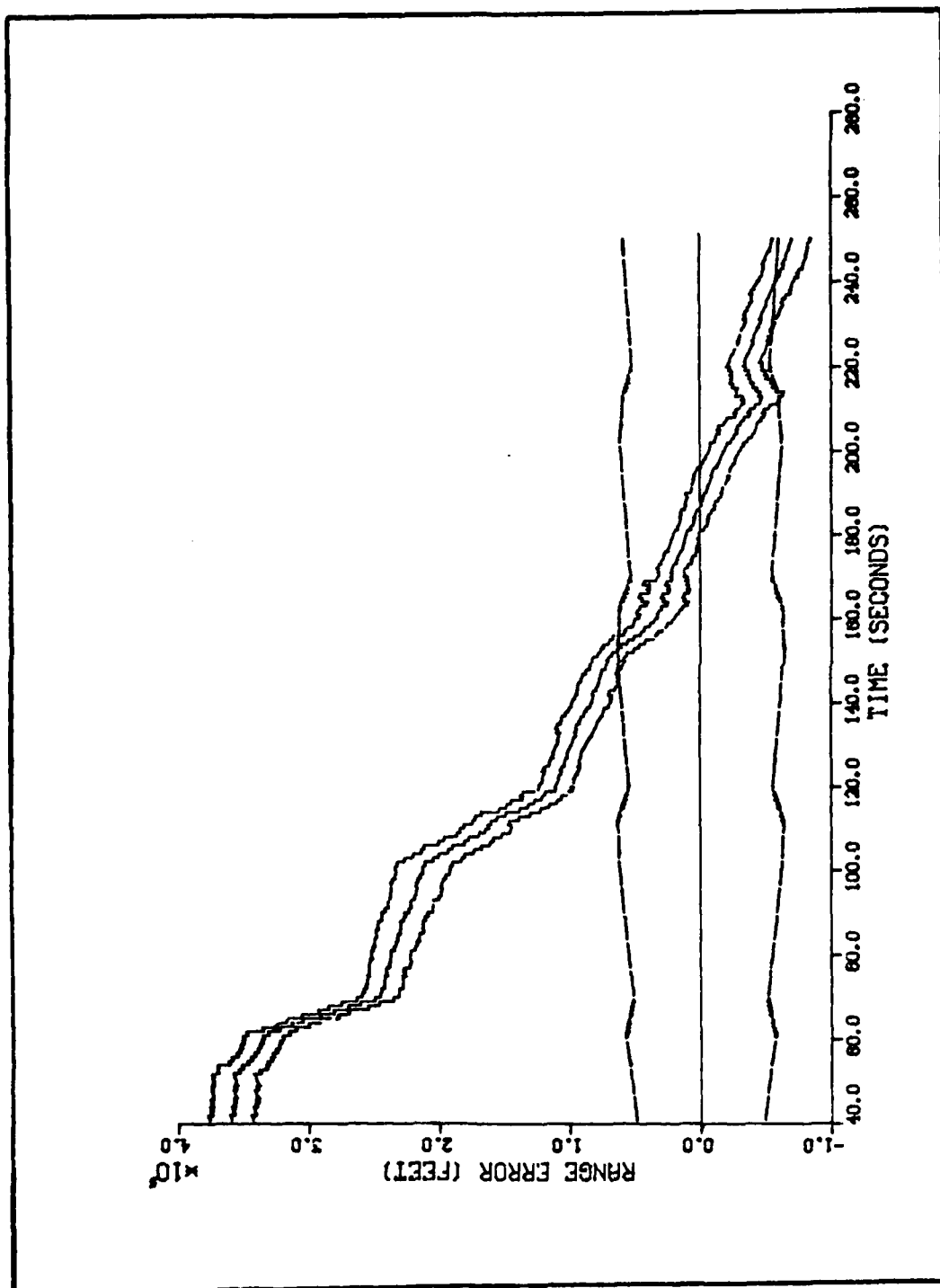


Figure L-27. Algorithm 5, Range Error Versus Time, Target Profile 8

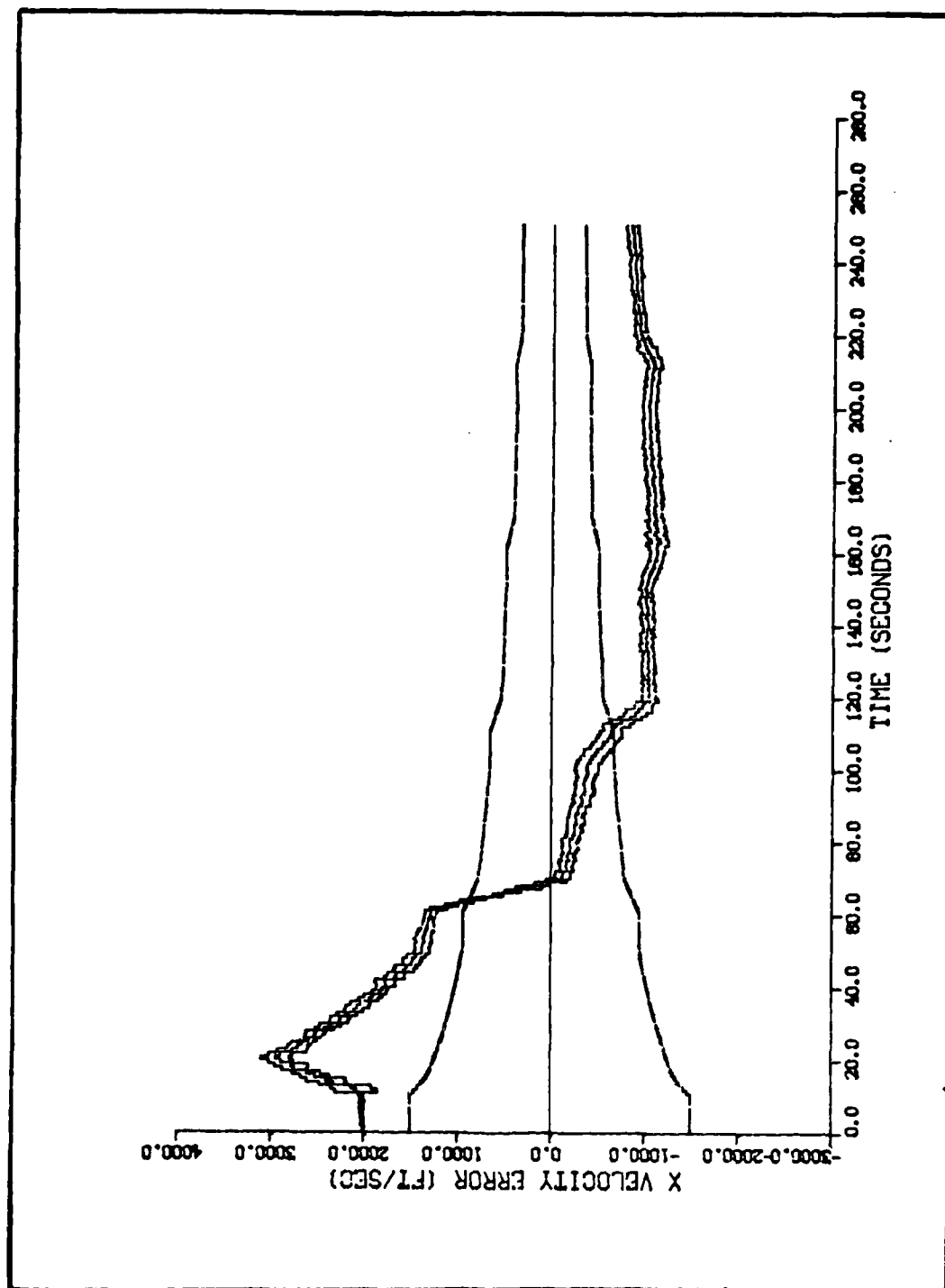


Figure L-28. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 8

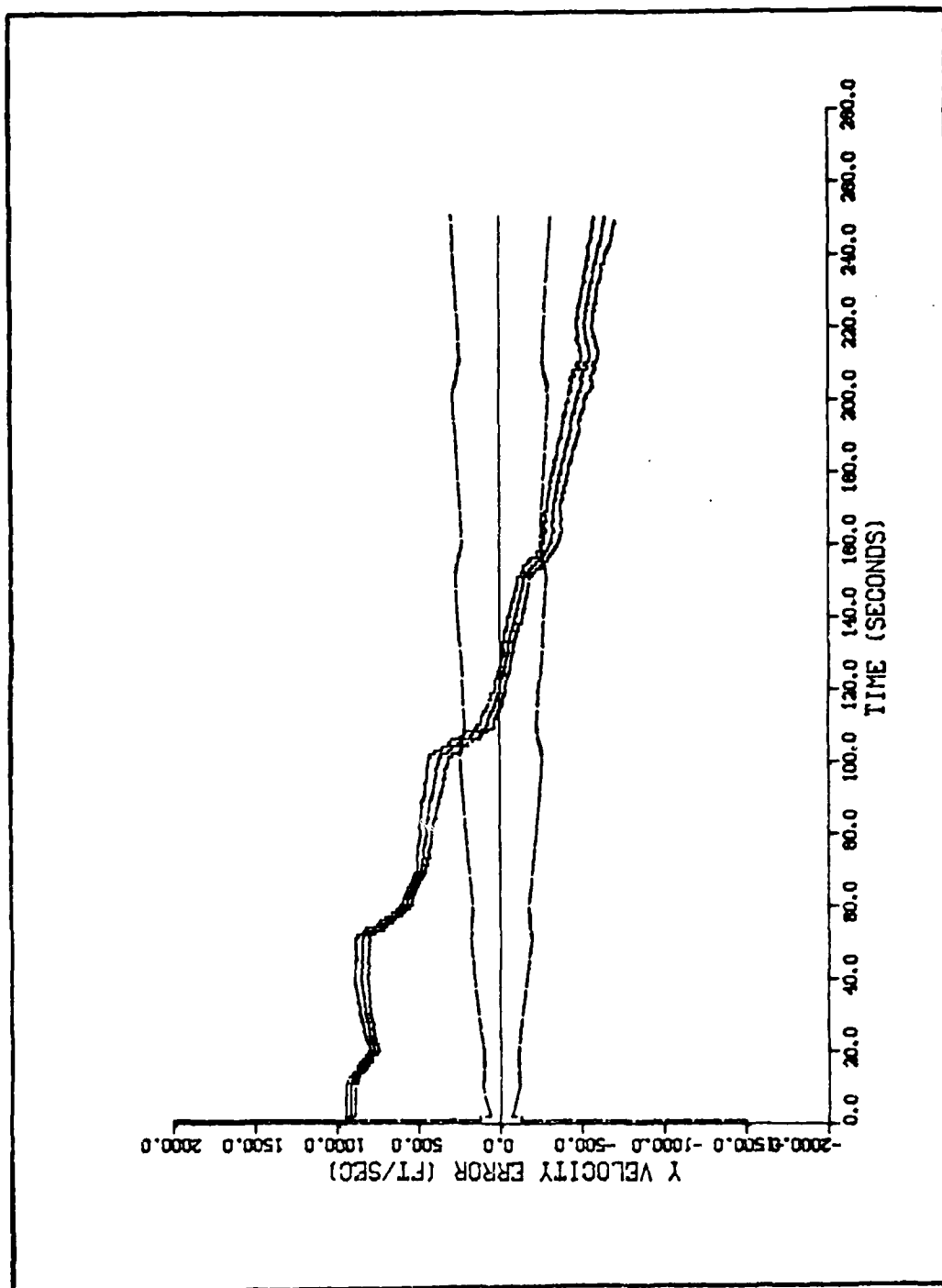


Figure L-29. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 8

APPENDIX M
GRAPHICAL RESULTS OF ALGORITHM SIX

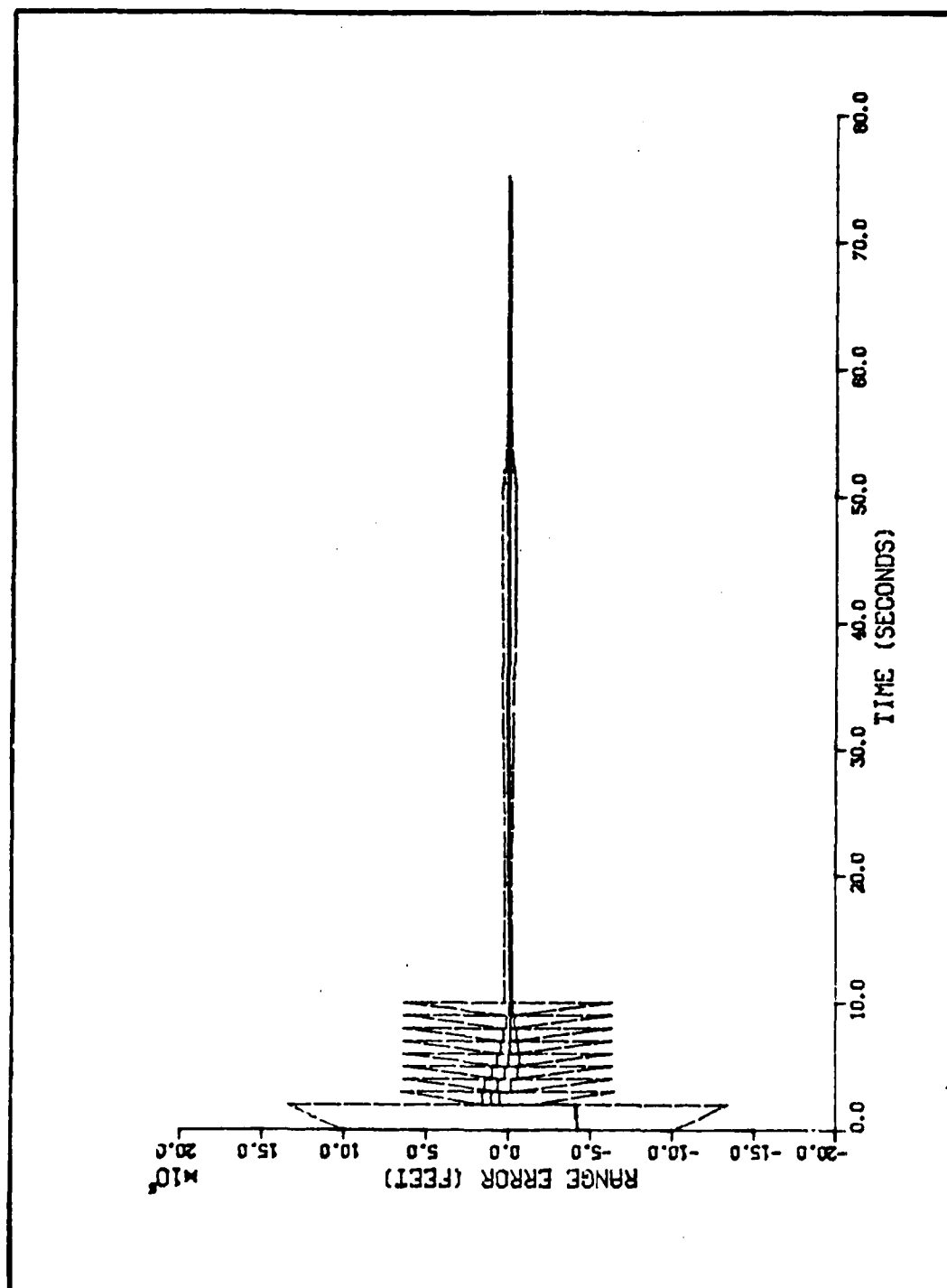


Figure M-1. Algorithm 6, Range Error Versus Time, Target Profile 1

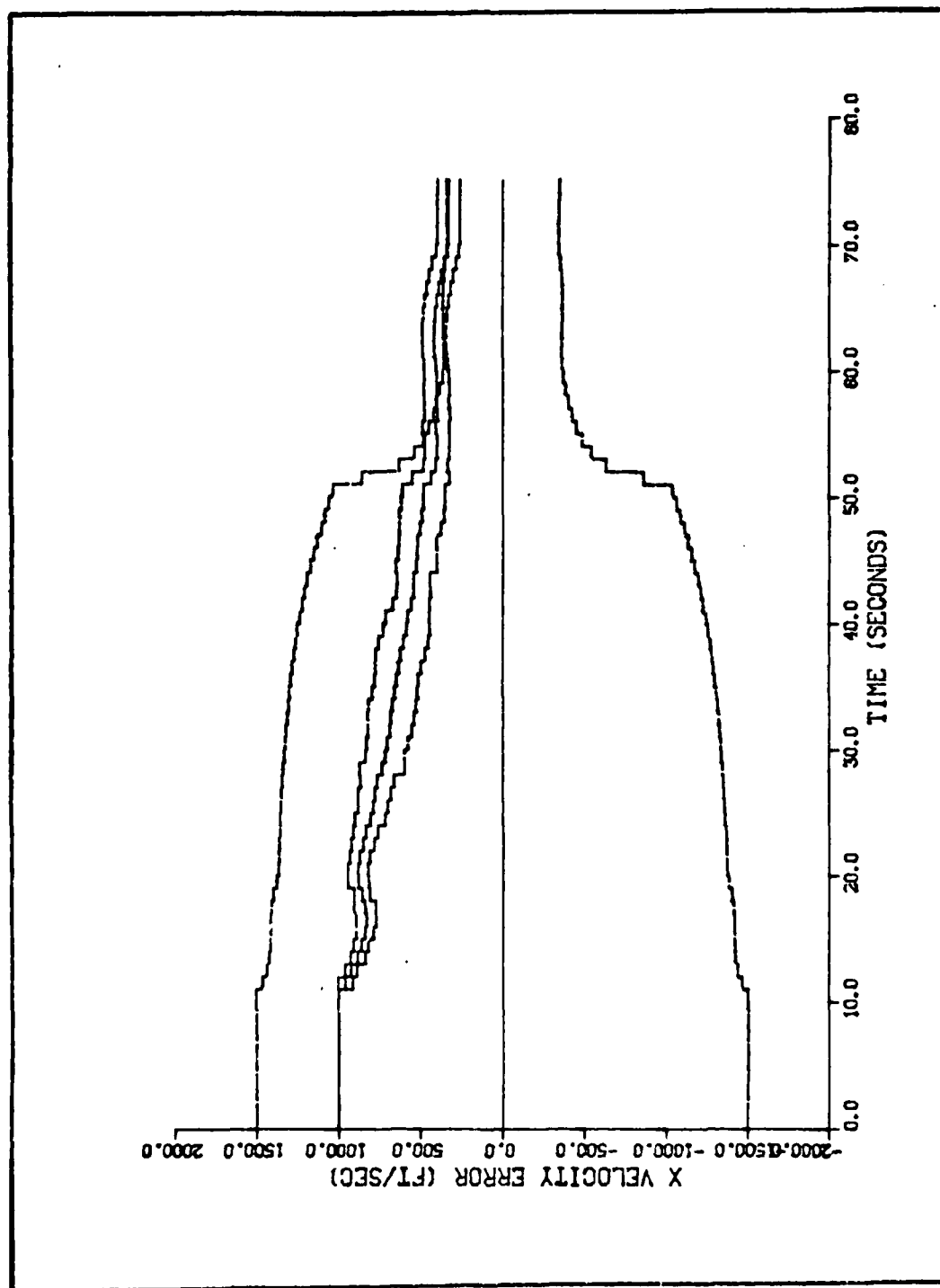


Figure M-2. Algorithm 6, Velocity Error in the X Direction Versus Time, Target Profile 1

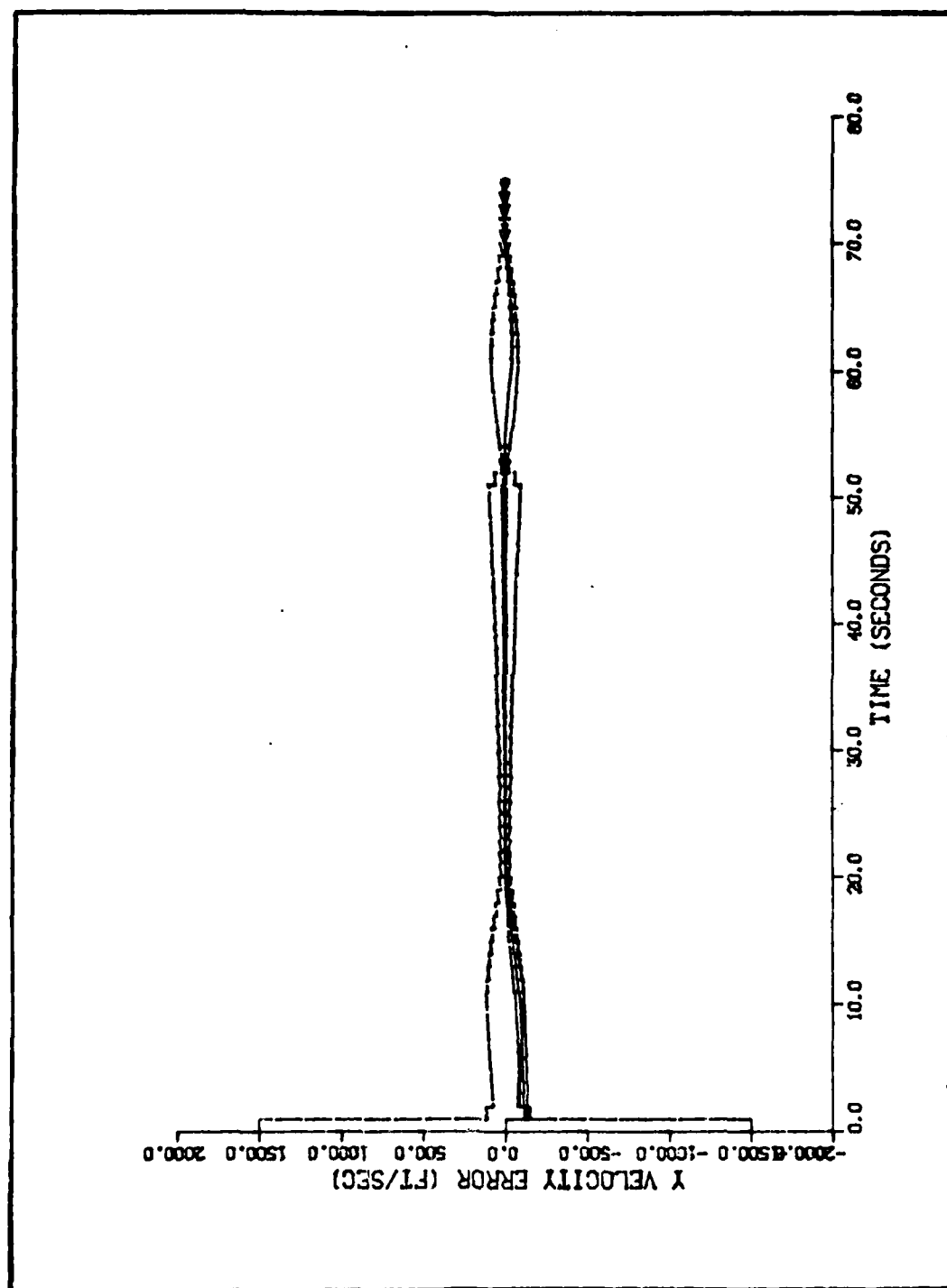


Figure M-3. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 1

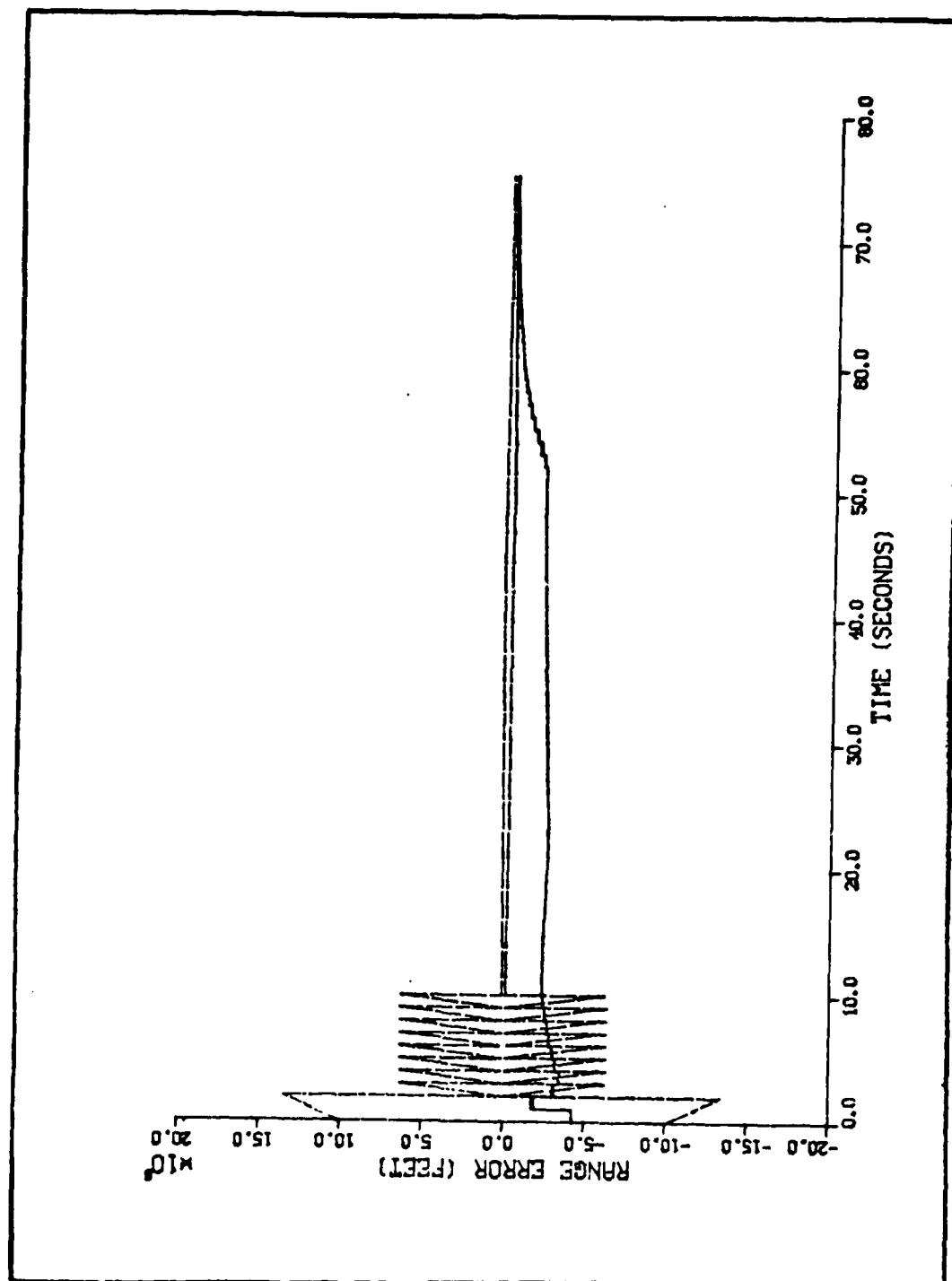


Figure M-4. Algorithm 6, Range Error Versus Time, Target Profile 2

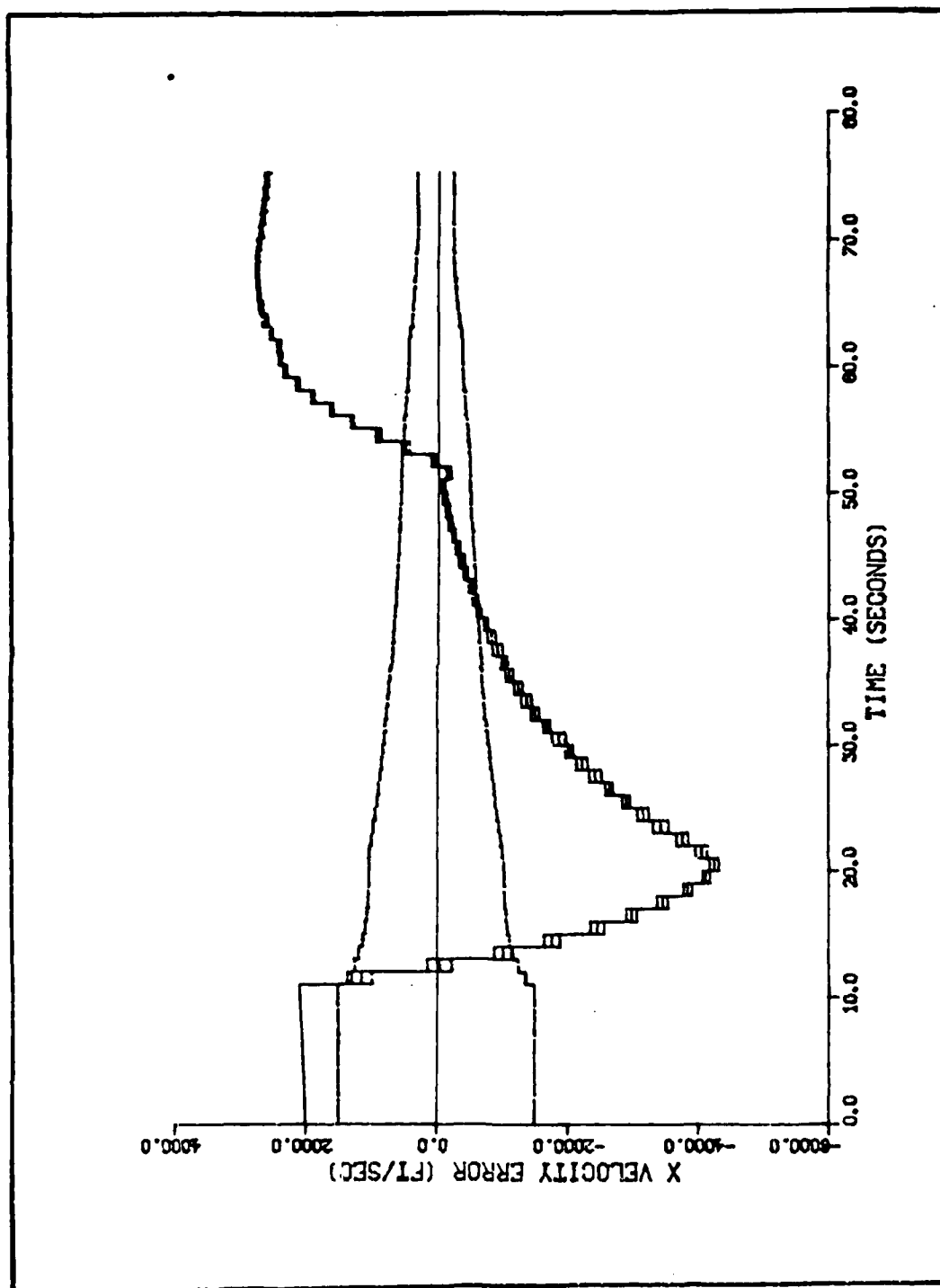


Figure M-5. Algorithm 6, Velocity Error in the X Direction Versus Time, Target Profile 2

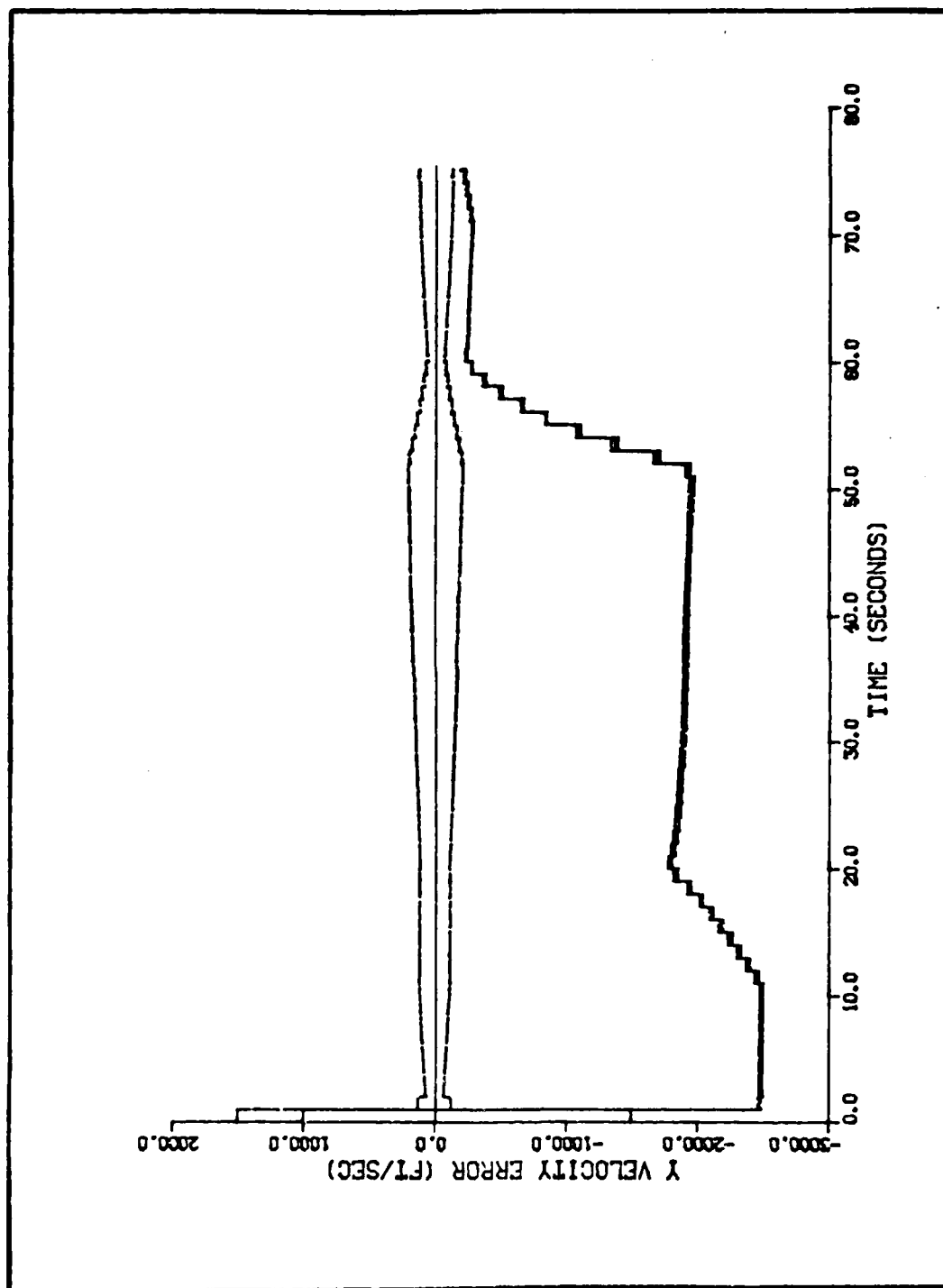


Figure M-6. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 2

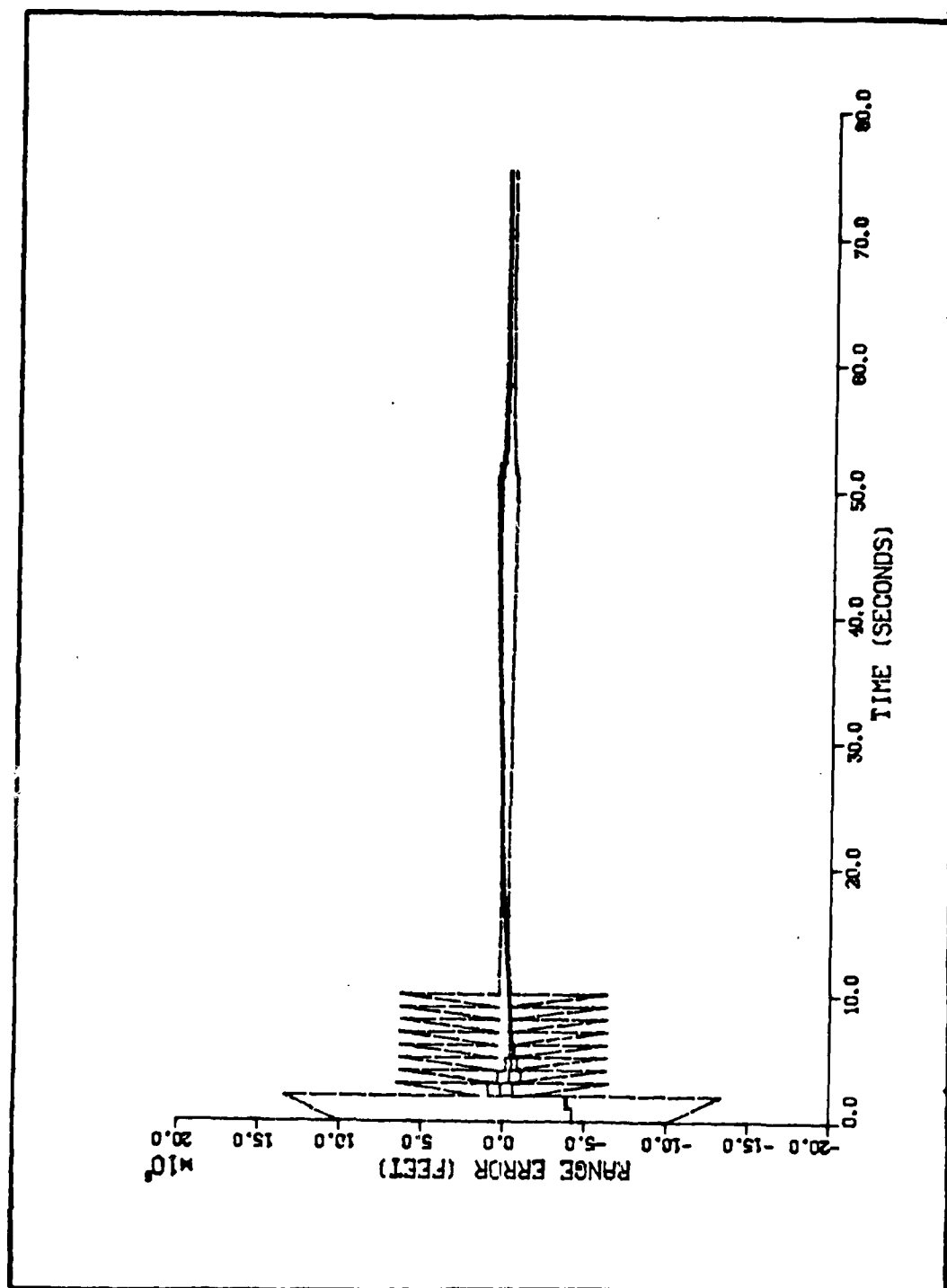


Figure M-7. Algorithm 6, Range Error Versus Time, Target Profile 3

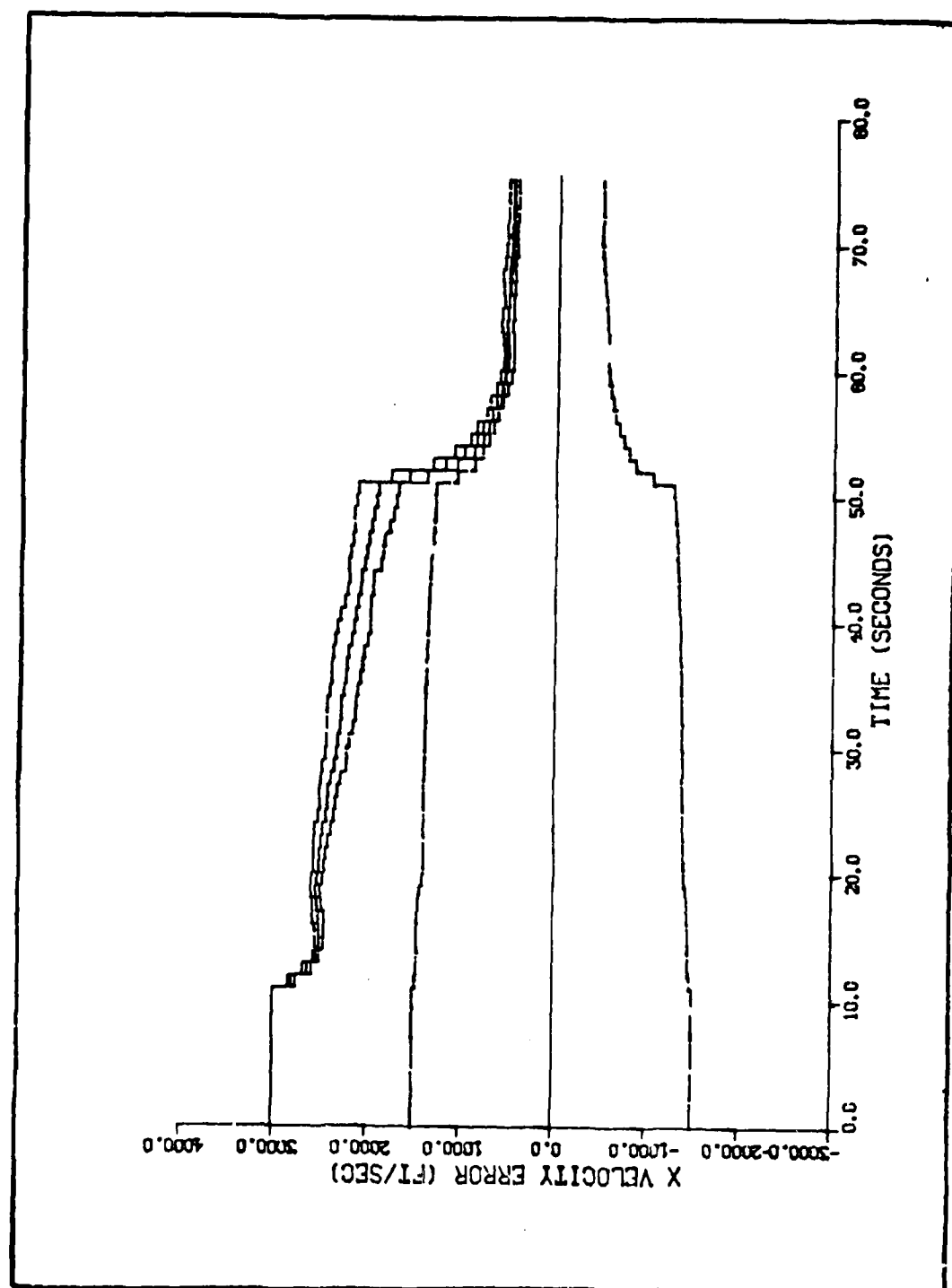


Figure M-8. Algorithm 6, Velocity Error in the X Direction Versus Time, Target Profile 3

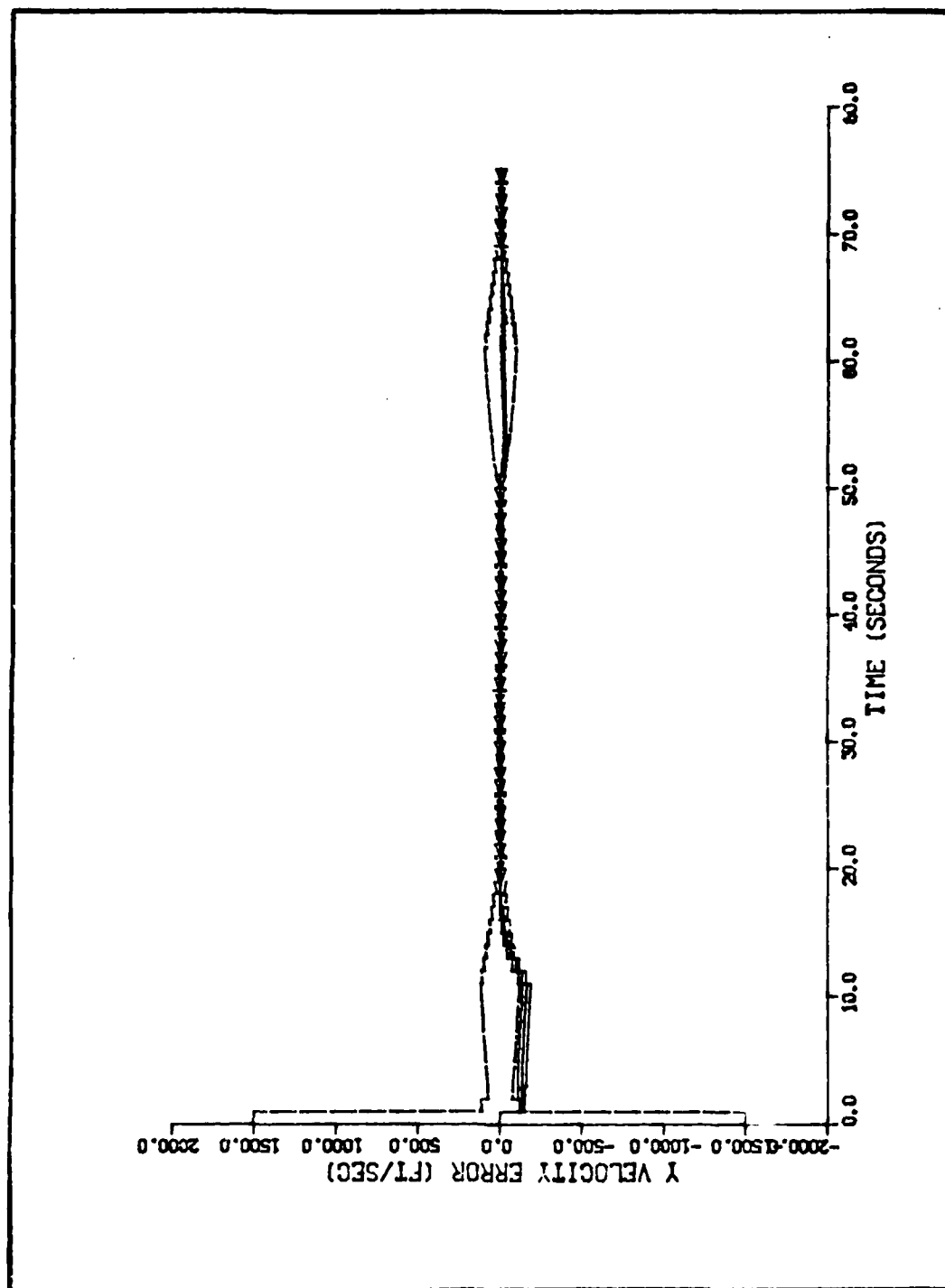


Figure M-9. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 3

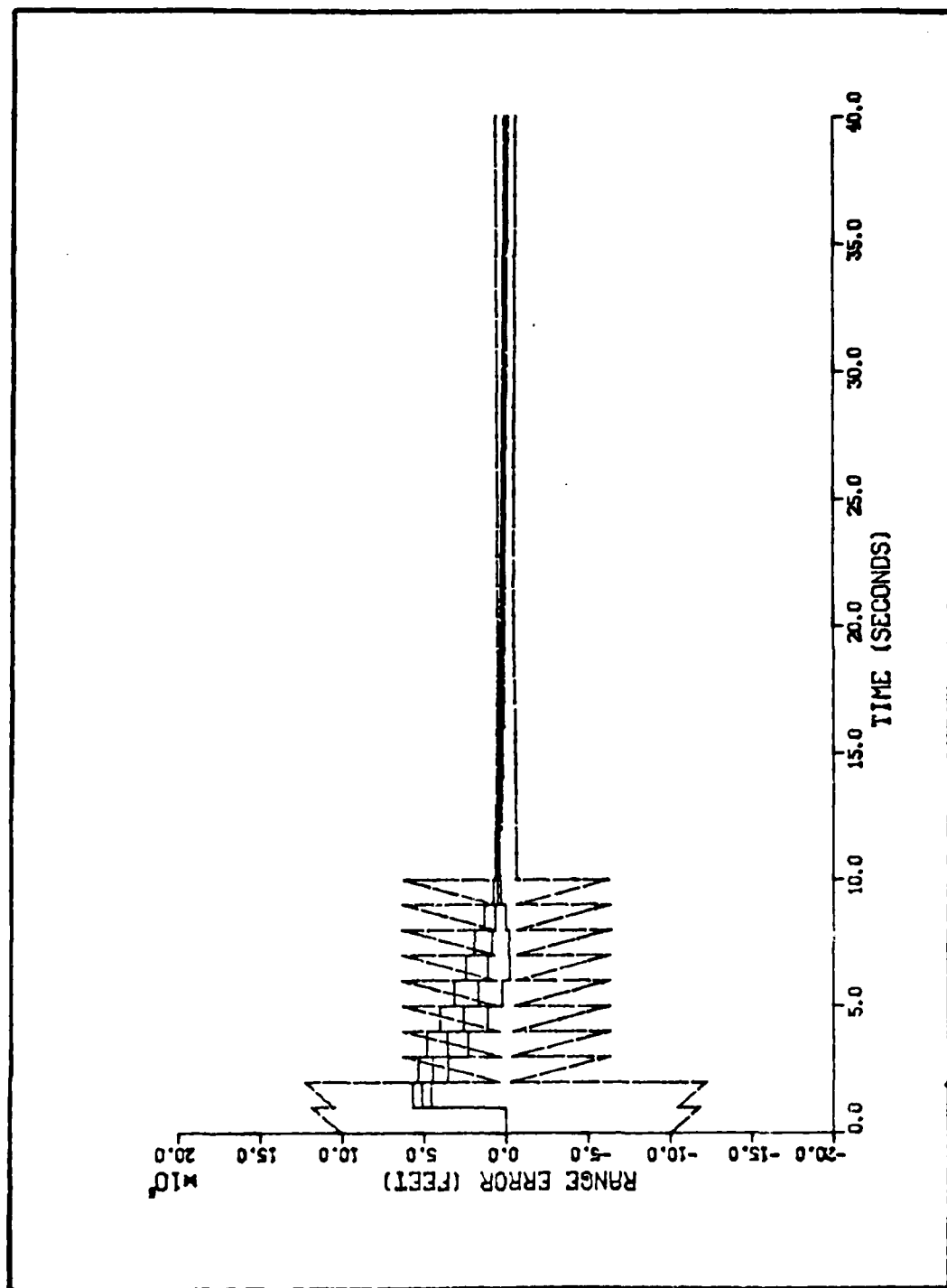


Figure M-10. Algorithm 6, Range Error Versus Time, Target Profile 4

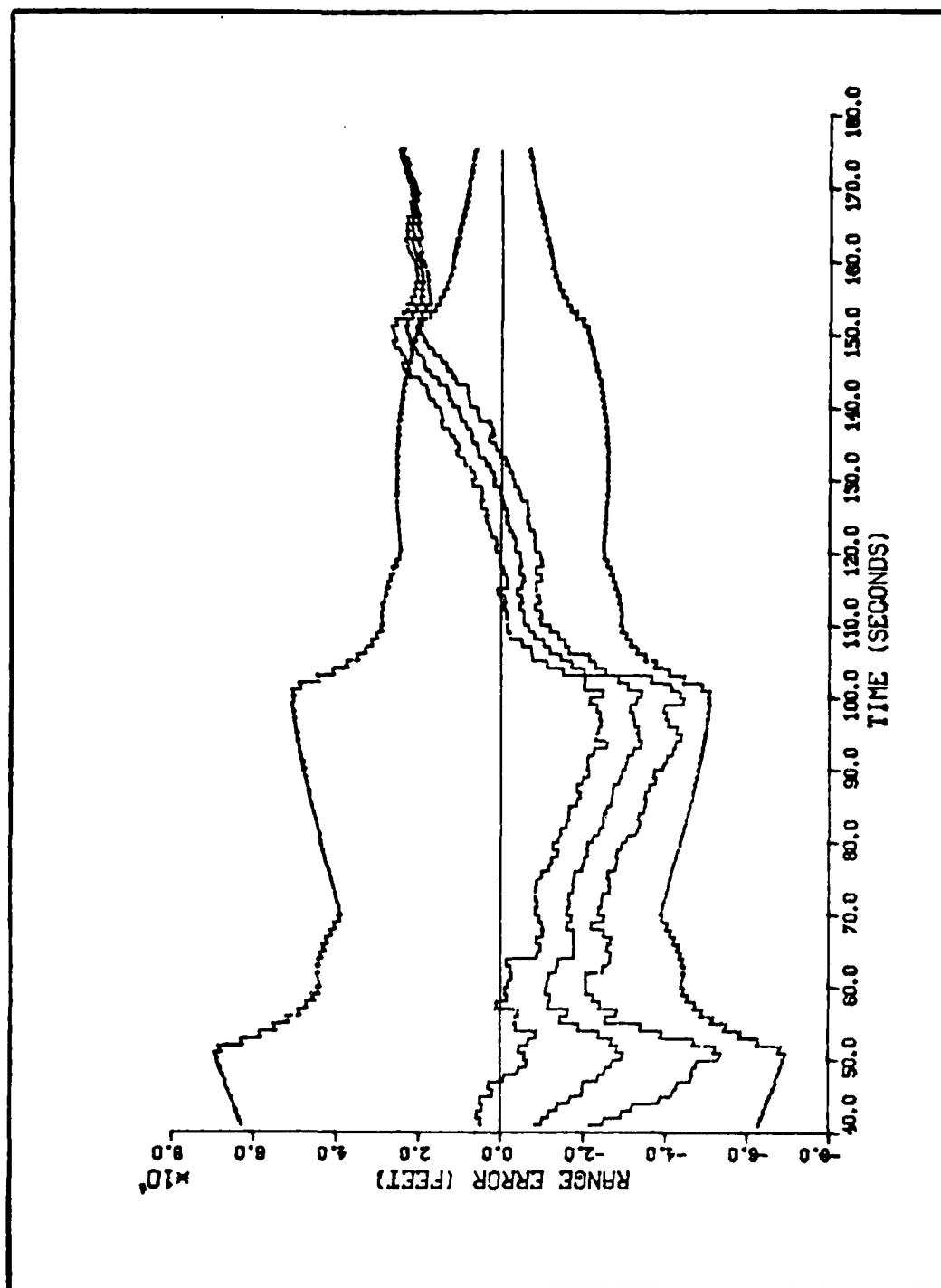


Figure M-11. Algorithm 6, Range Error Versus Time, Target Profile 4

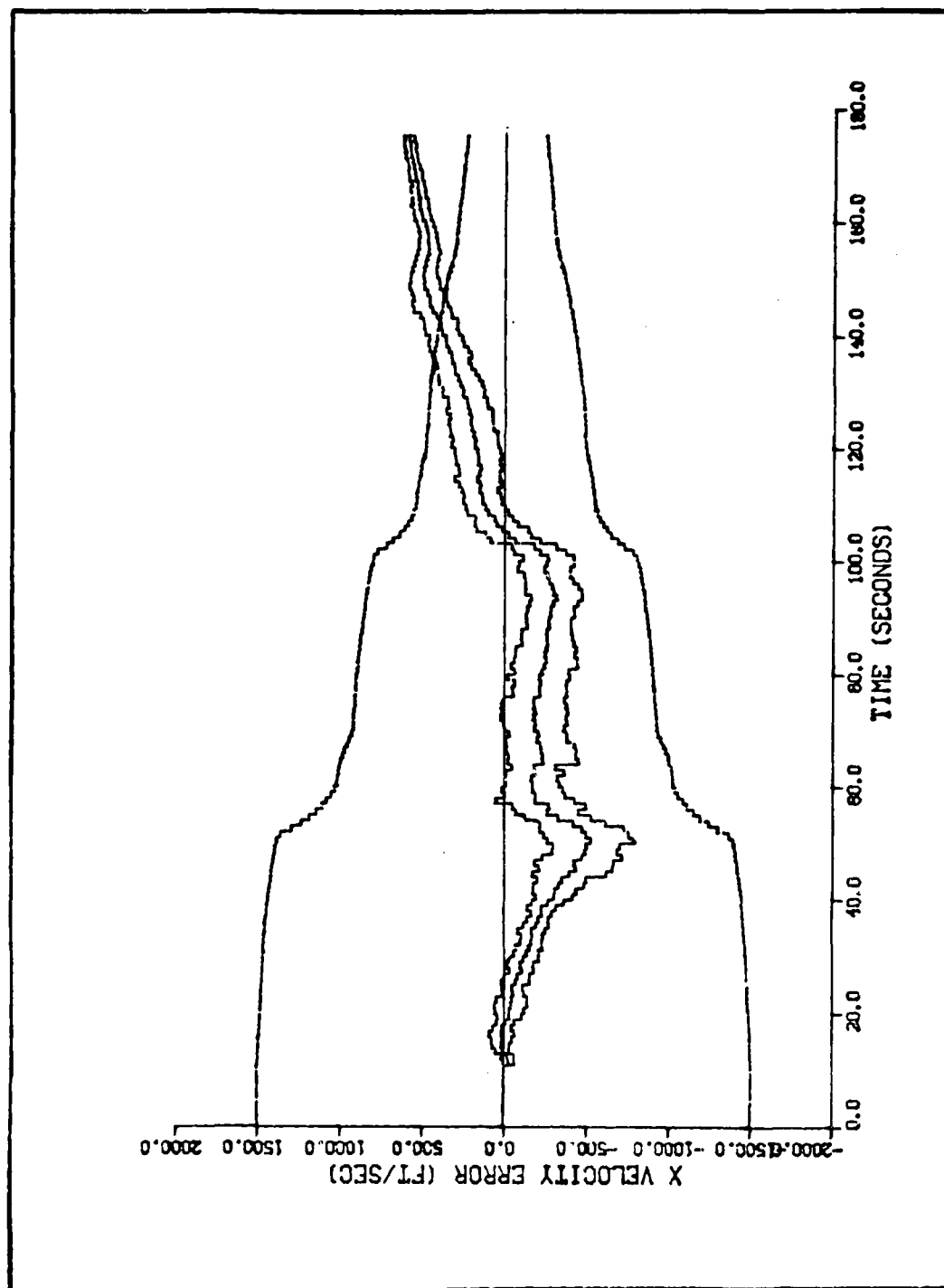


Figure M-12. Algorithm 6, Velocity Error in the X Direction Versus Time, Target Profile 4

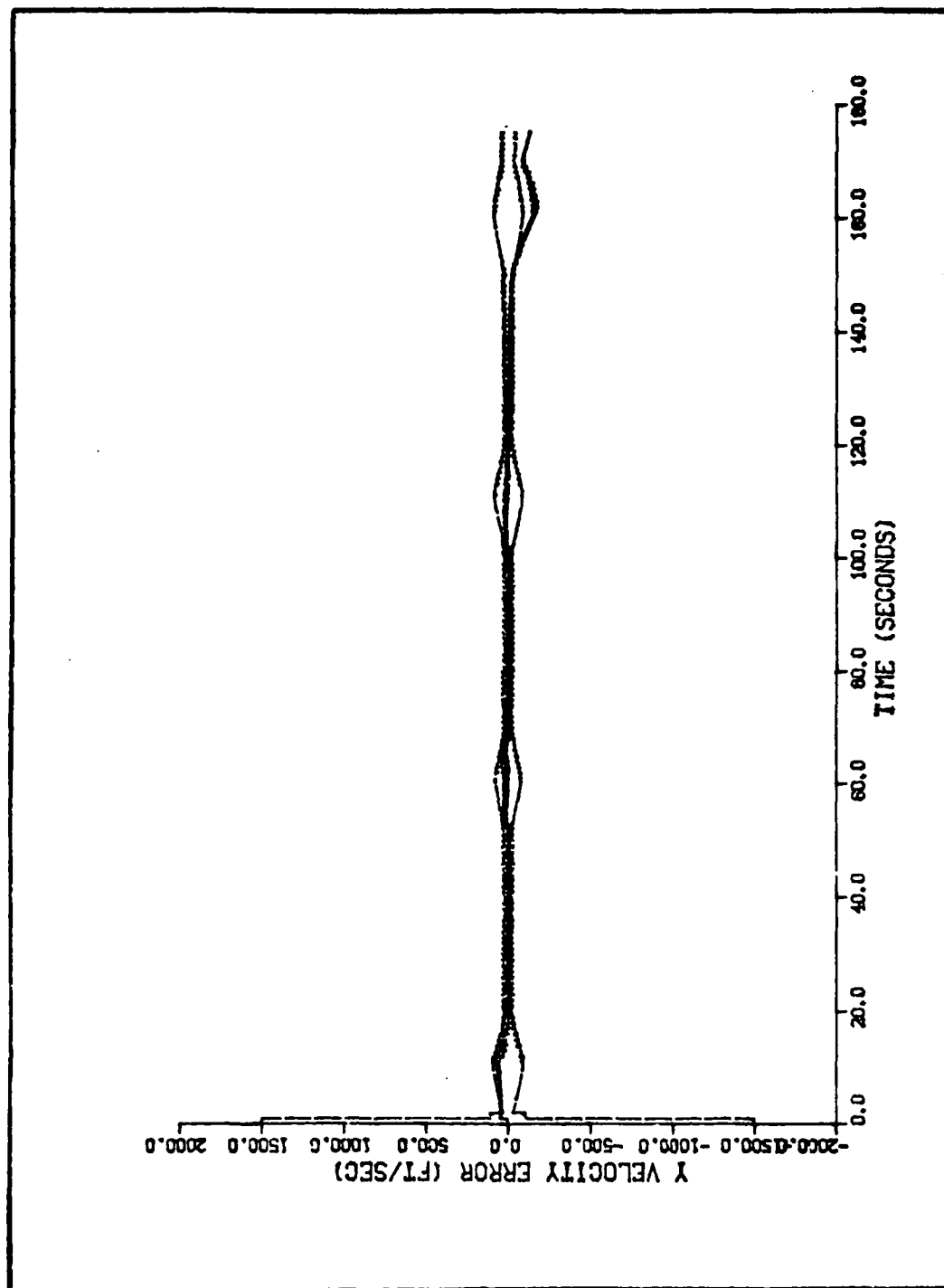


Figure M-13. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 4

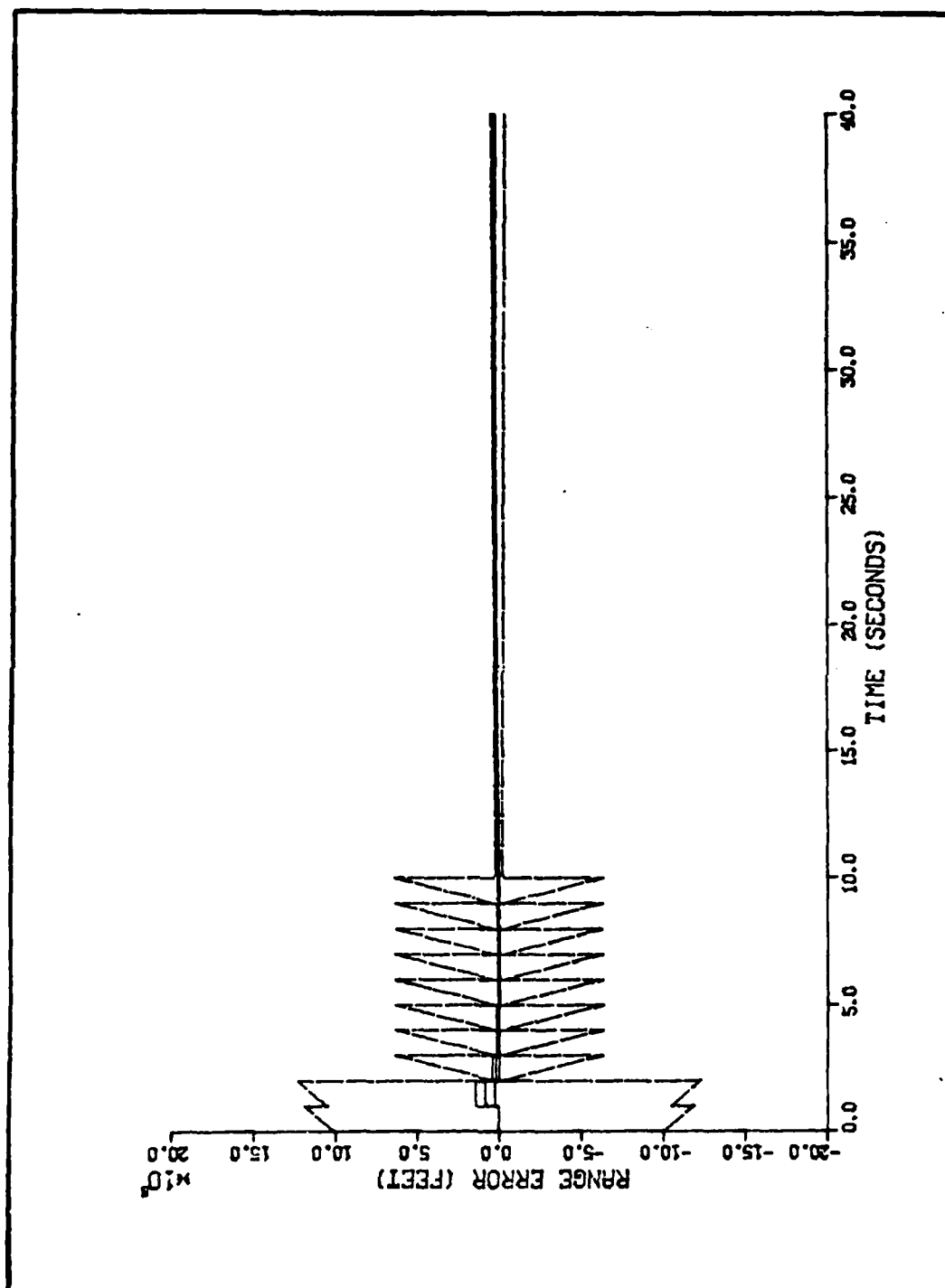


Figure M-14. Algorithm 6, Range Error Versus Time, Target Profile 5

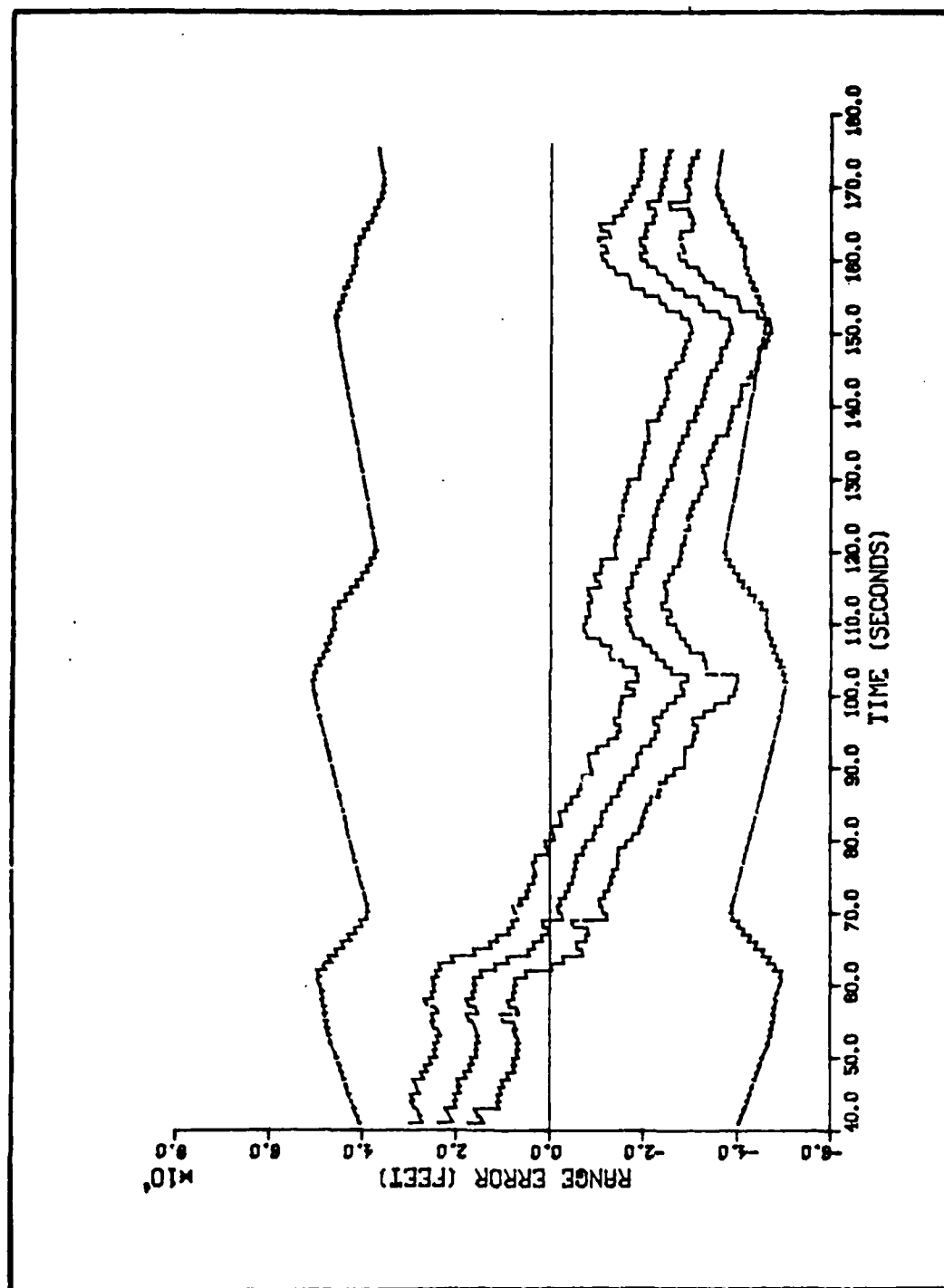


Figure M-15. Algorithm 6, Range Error Versus Time, Target Profile 5

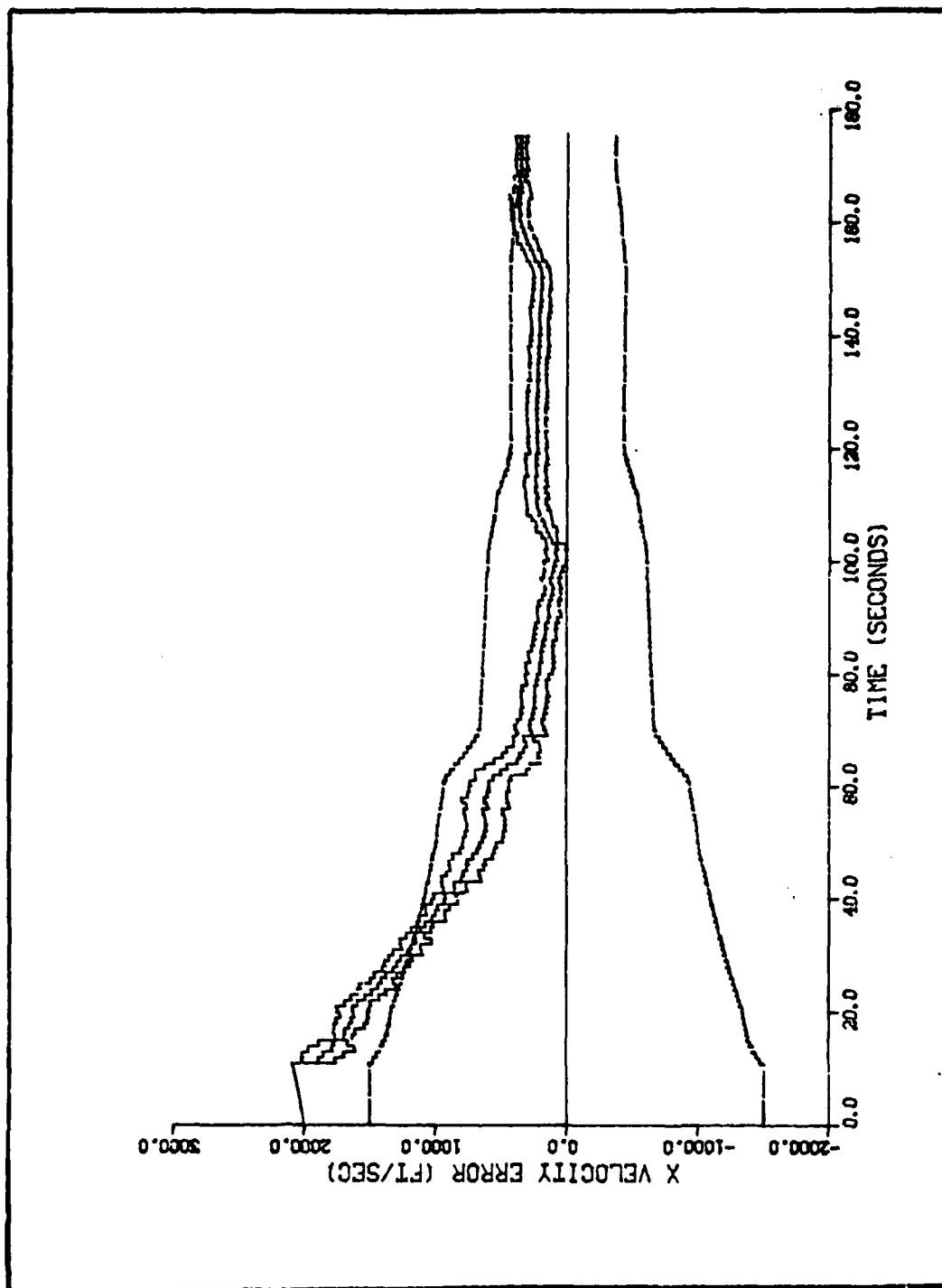


Figure M-16. Algorithm 6, Velocity Error in the X Direction Versus Time, Target Profile 5

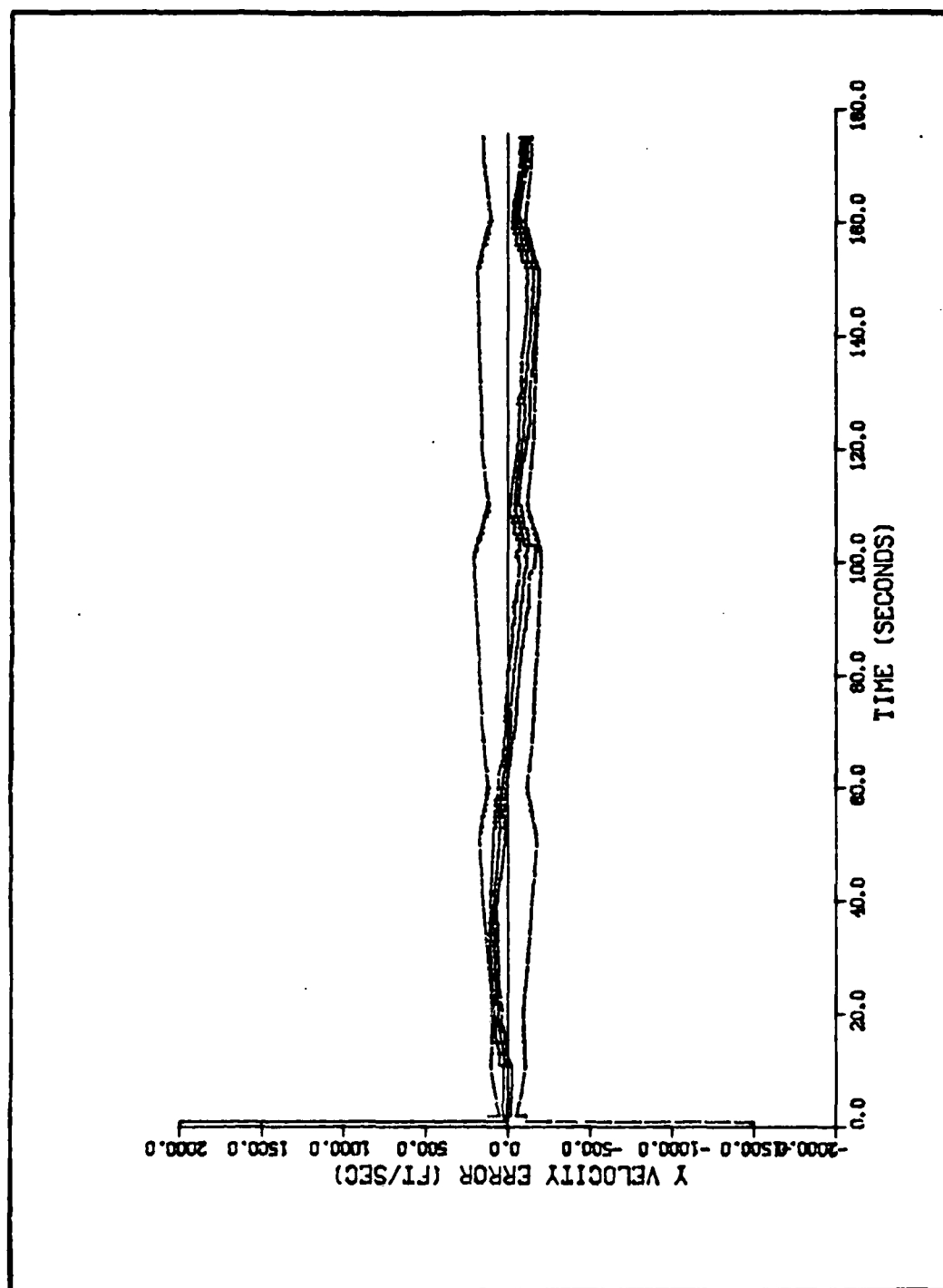


Figure M-17. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 5

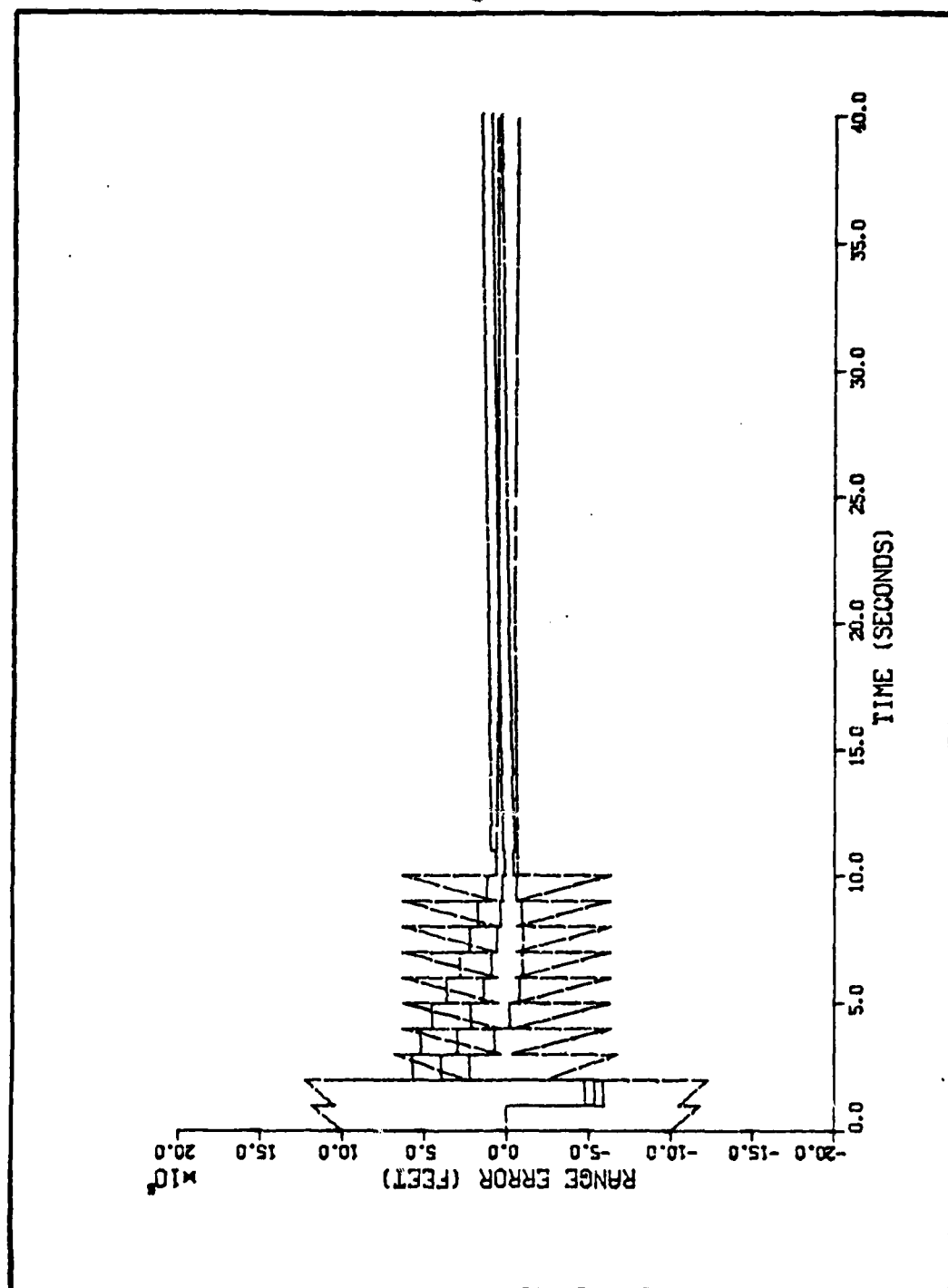


Figure M-18. Algorithm 6, Range Error Versus Time, Target Profile 6

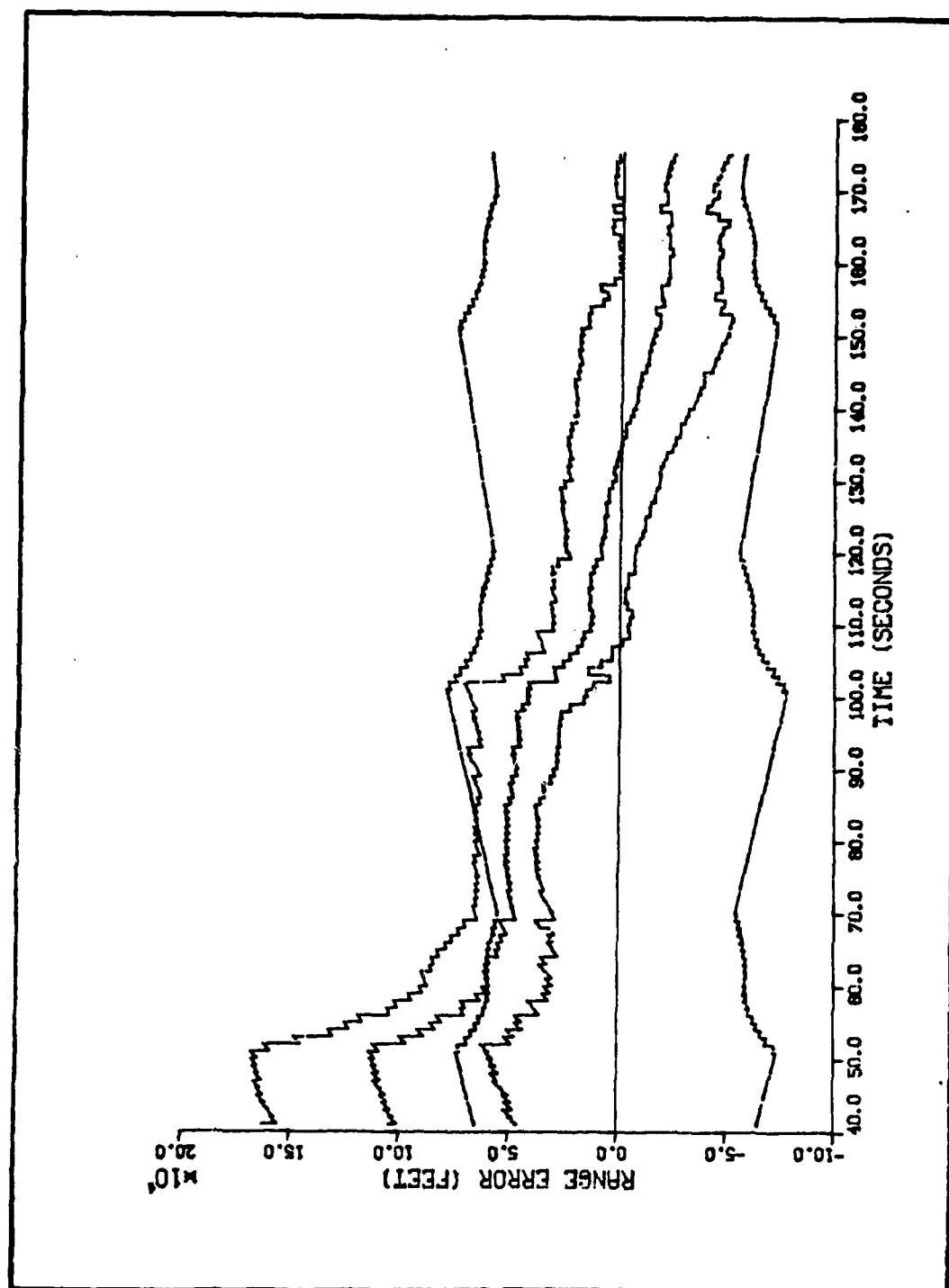


Figure M-19. Algorithm 6, Range Error Versus Time, Target Profile 6

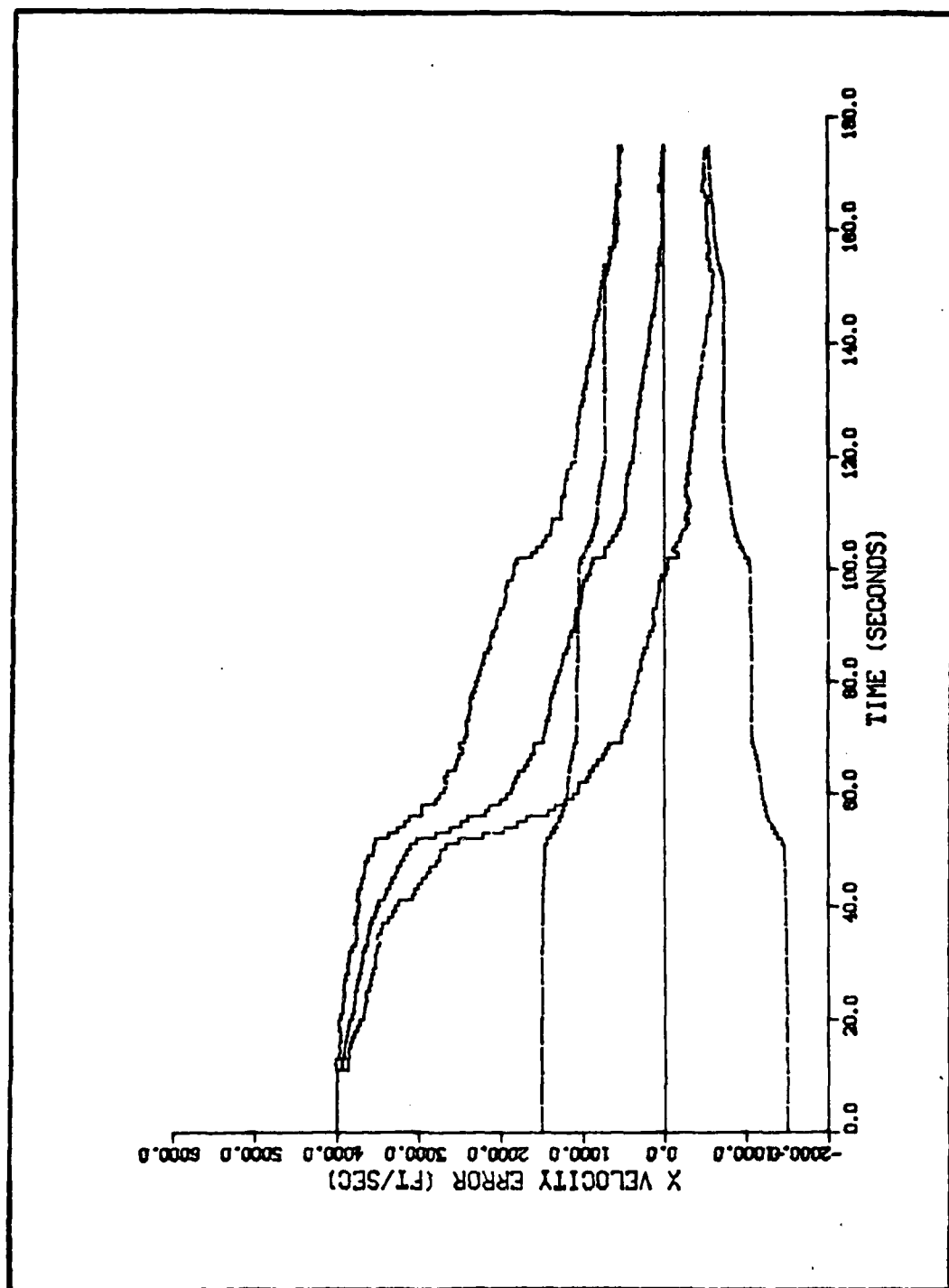


Figure M-20. Algorithm 6, Velocity Error in the X Direction Versus Time, Target Profile 6

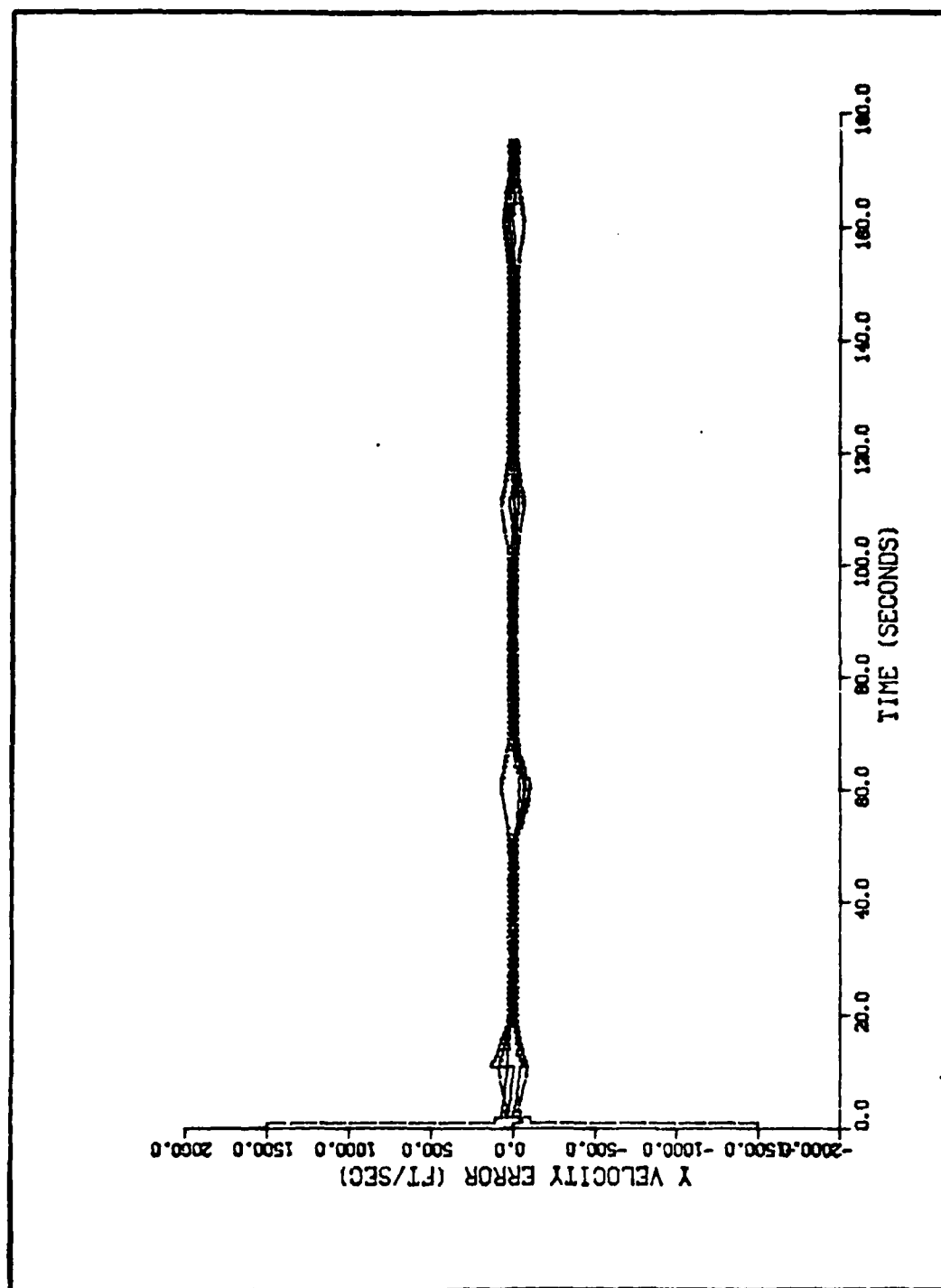


Figure M-21. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 6

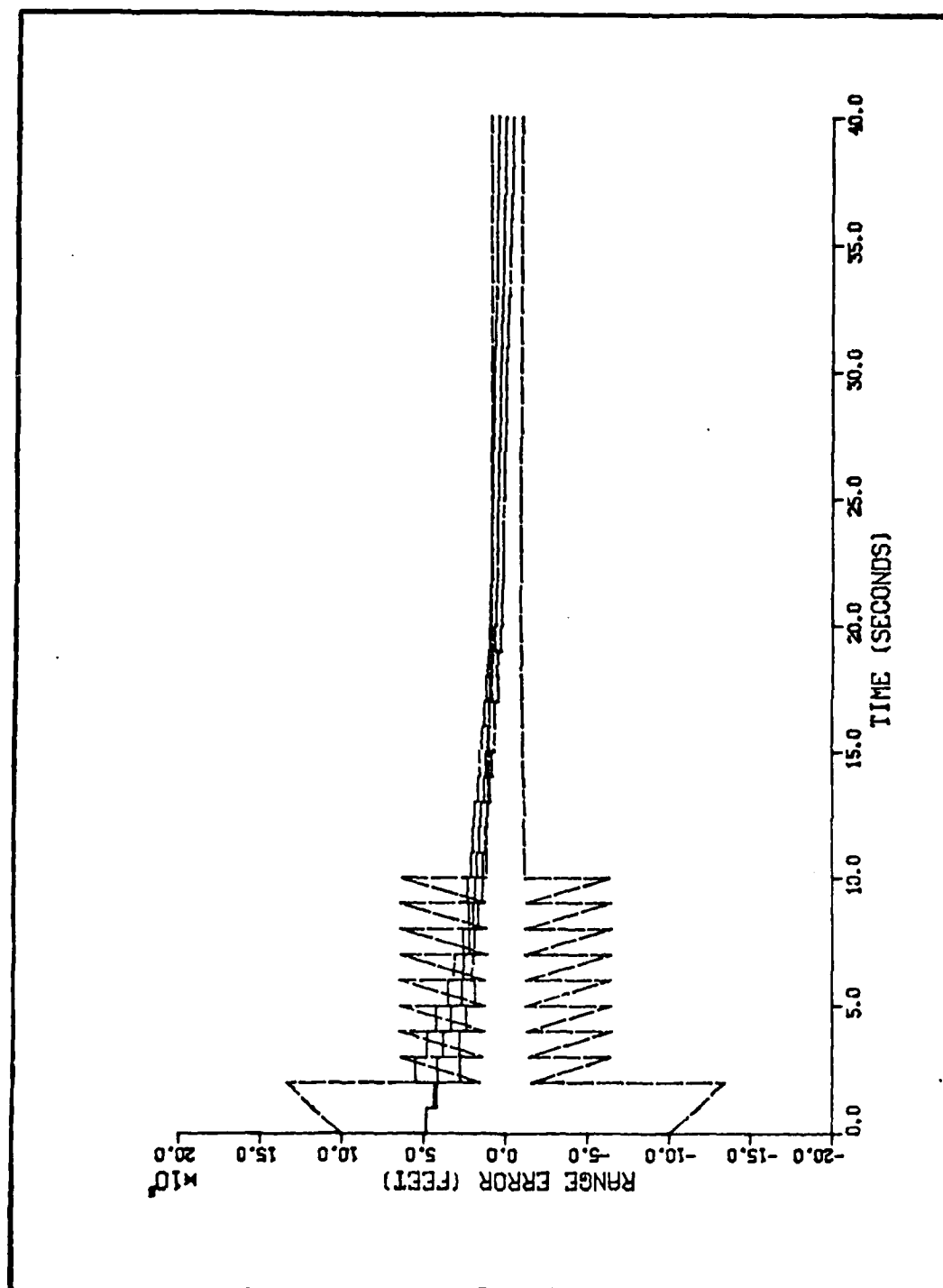


Figure M-22. Algorithm 6, Range Error Versus Time, Target Profile 7

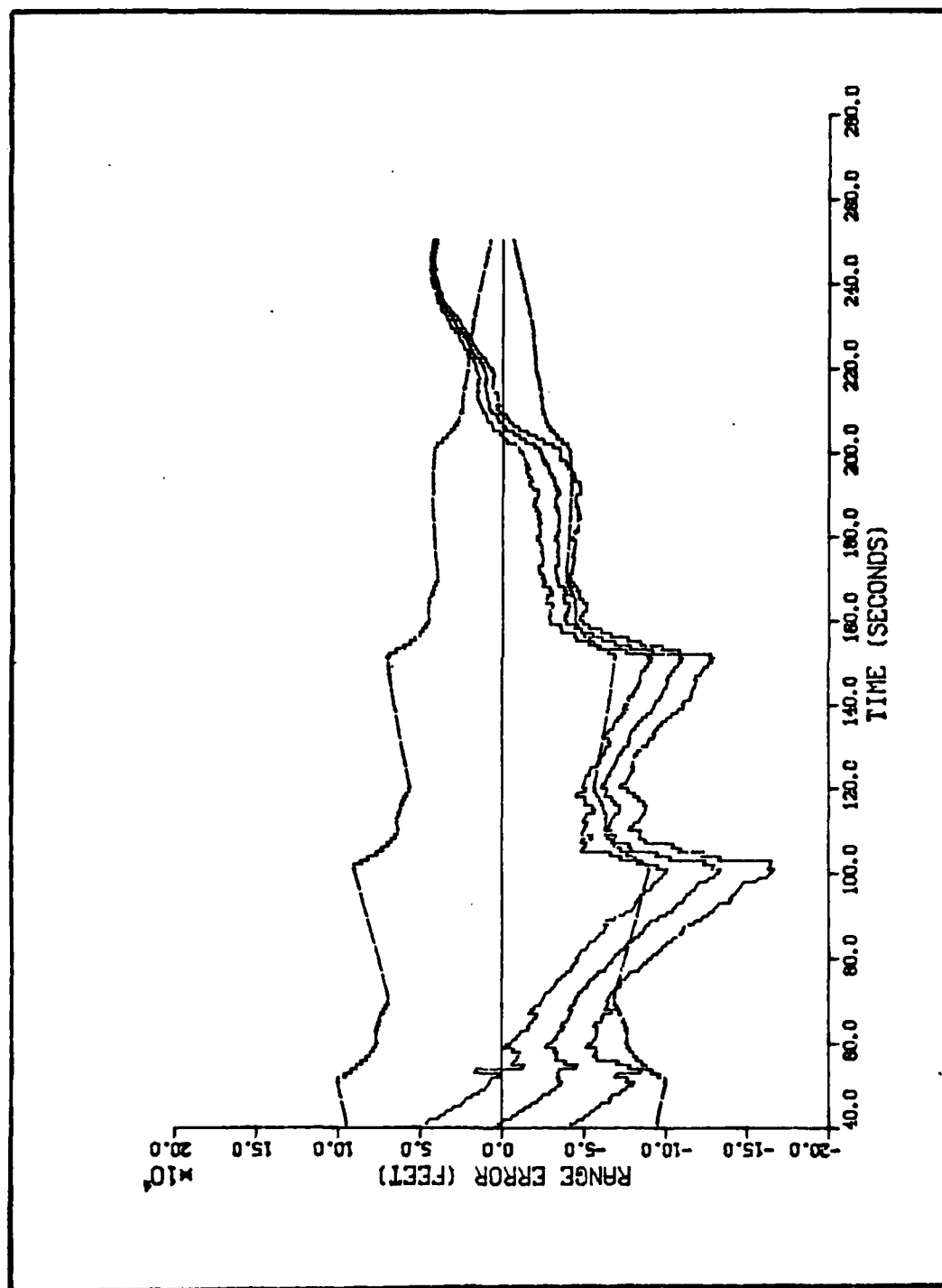


Figure M-23. Algorithm 6, Range Error Versus Time, T ... Profile 7

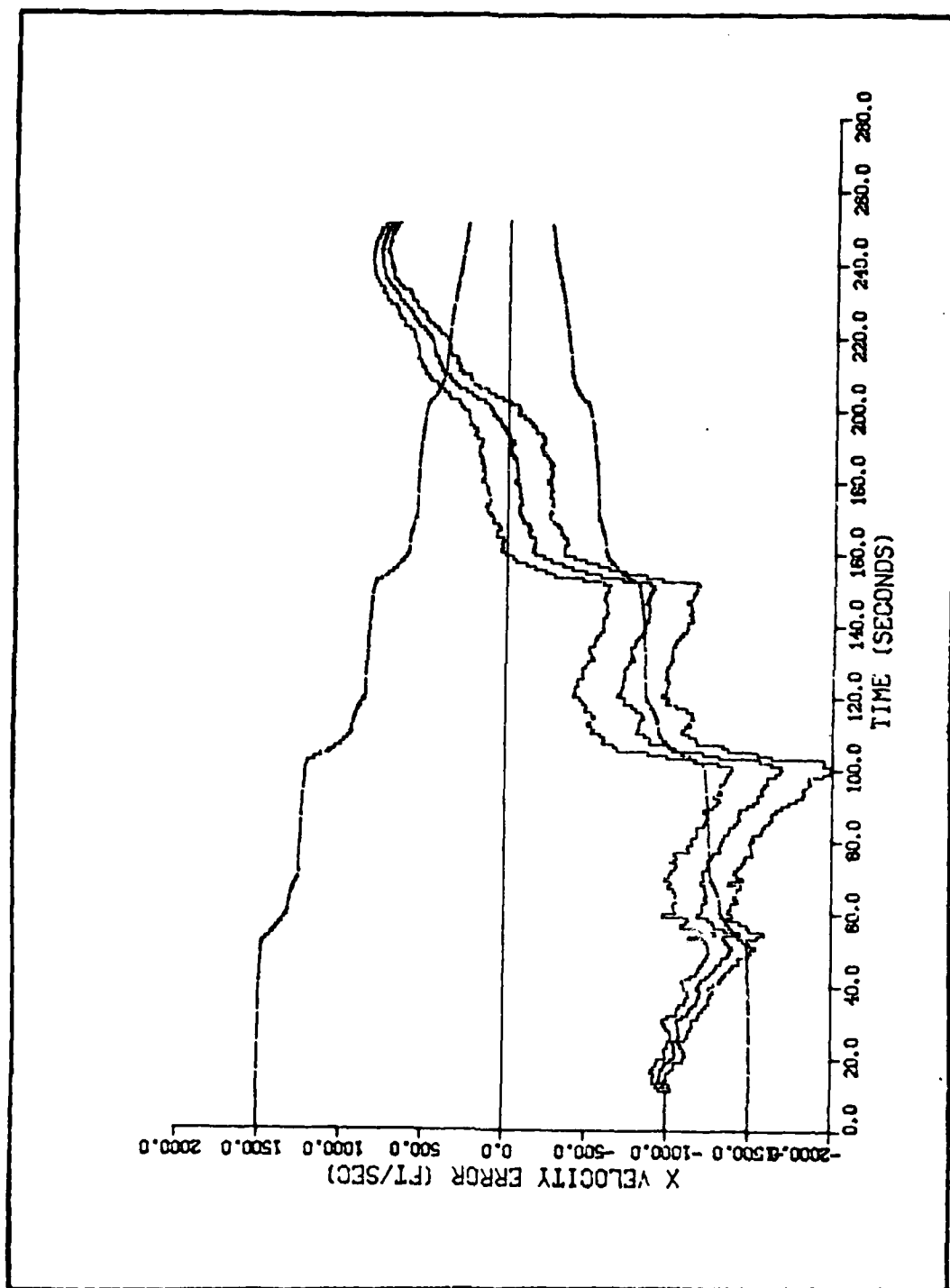


Figure M-24. Algorithm 6, Velocity Error in the X Direction Versus Time,
Target Profile 7

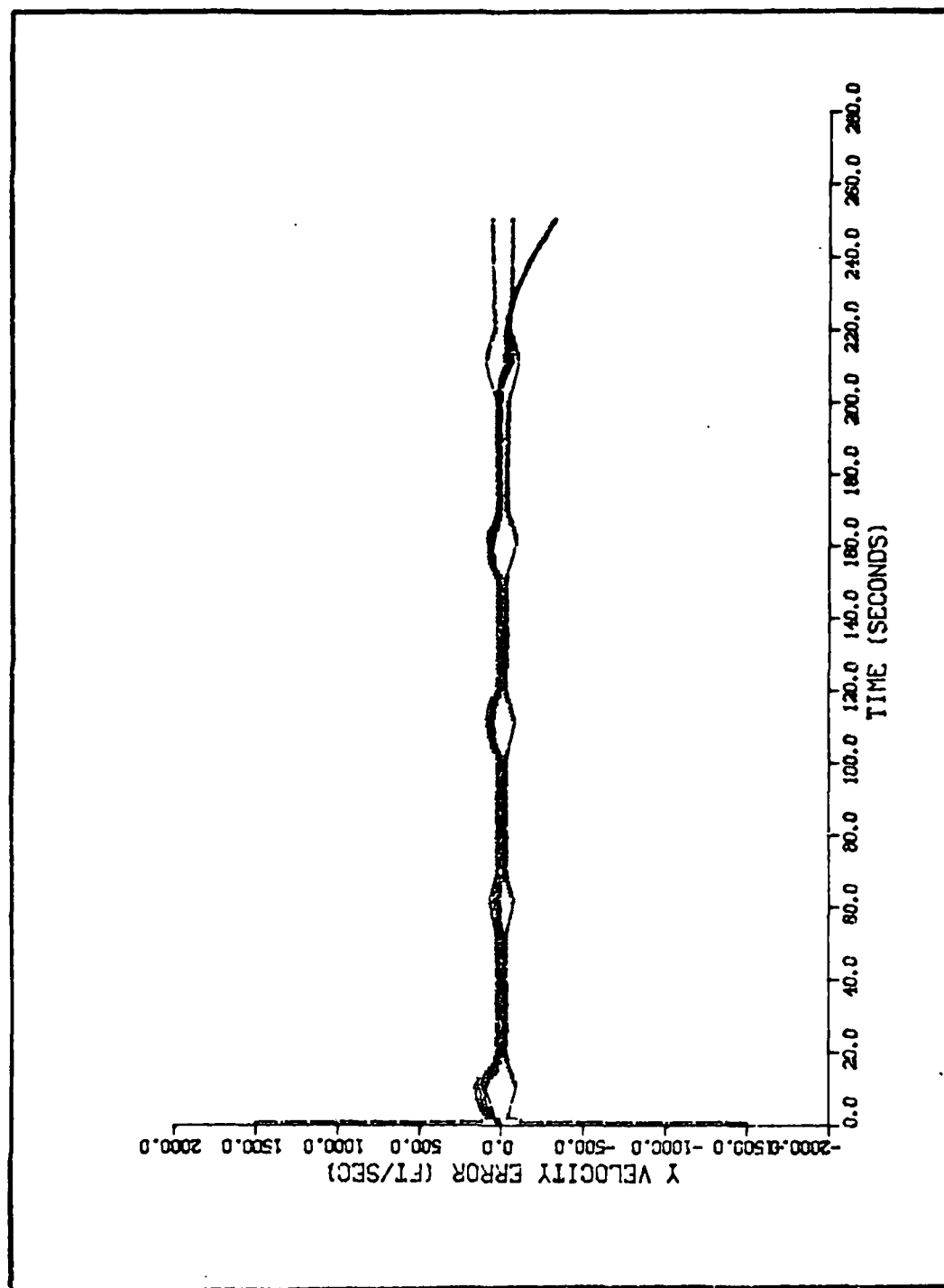


Figure M-25. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 7

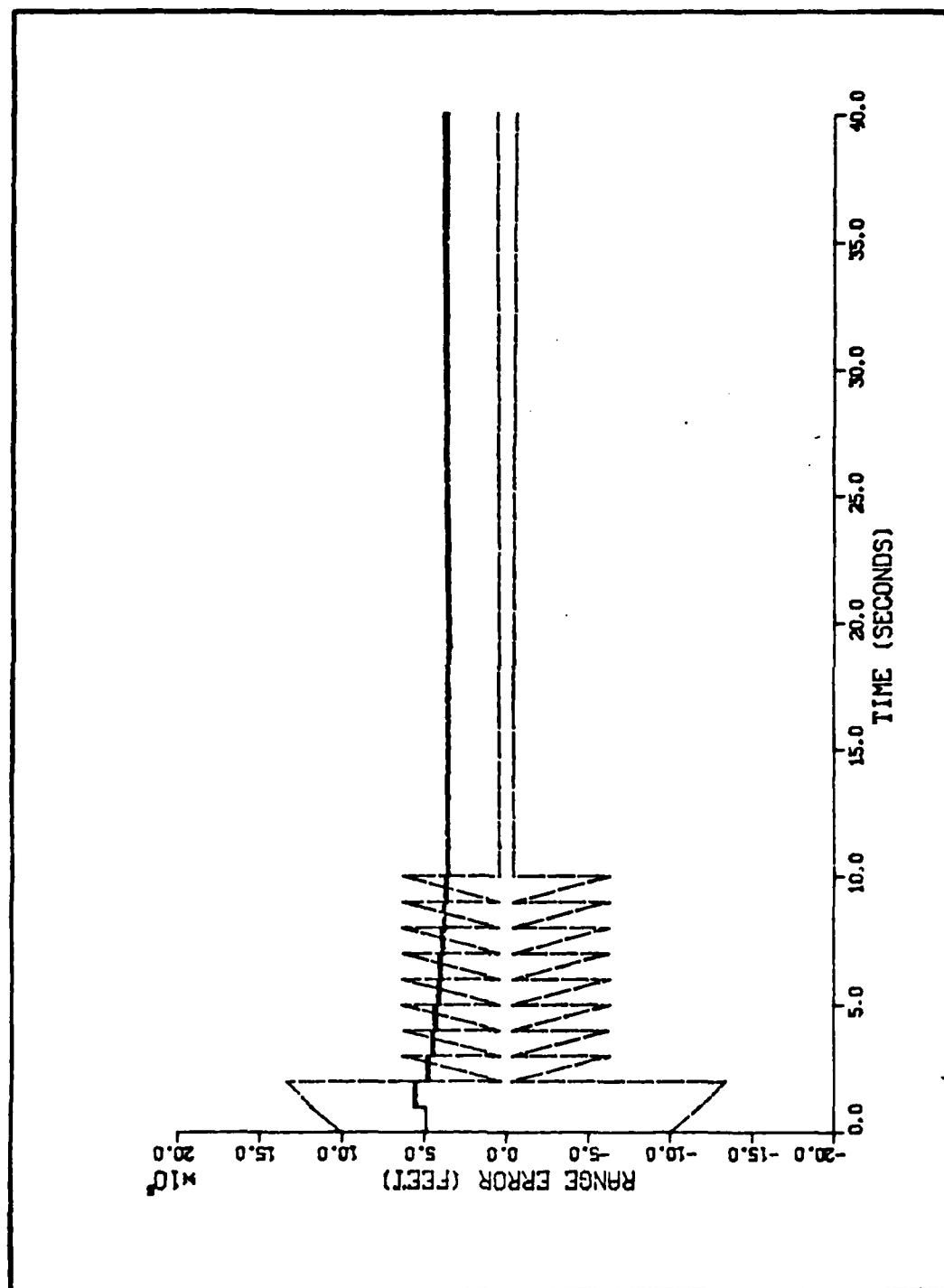


Figure M-26. Algorithm 6, Range Error Versus Time, Target Profile 8

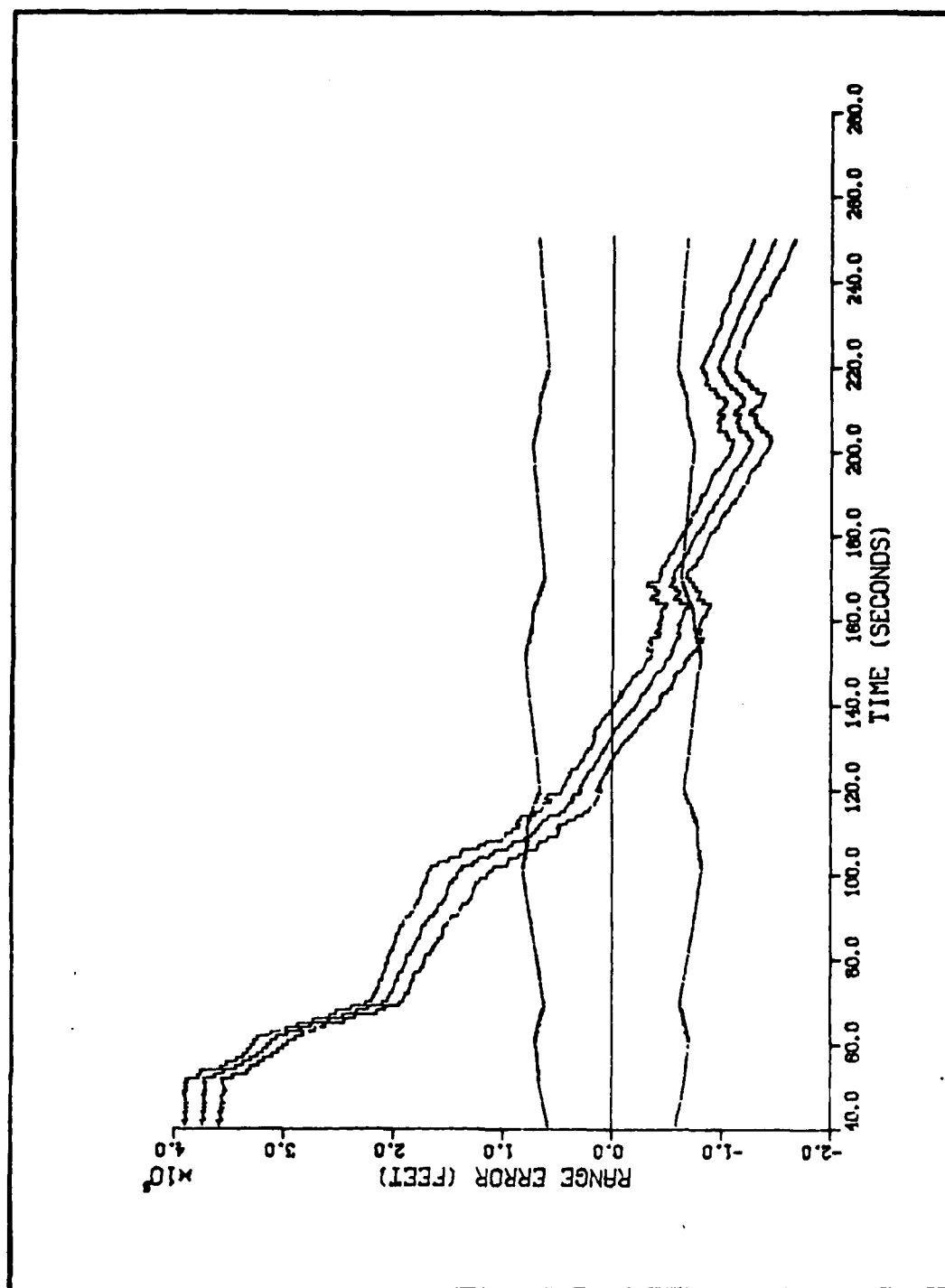


Figure M-27. Algorithm 6, Range Error Versus Time, Target Profile 8

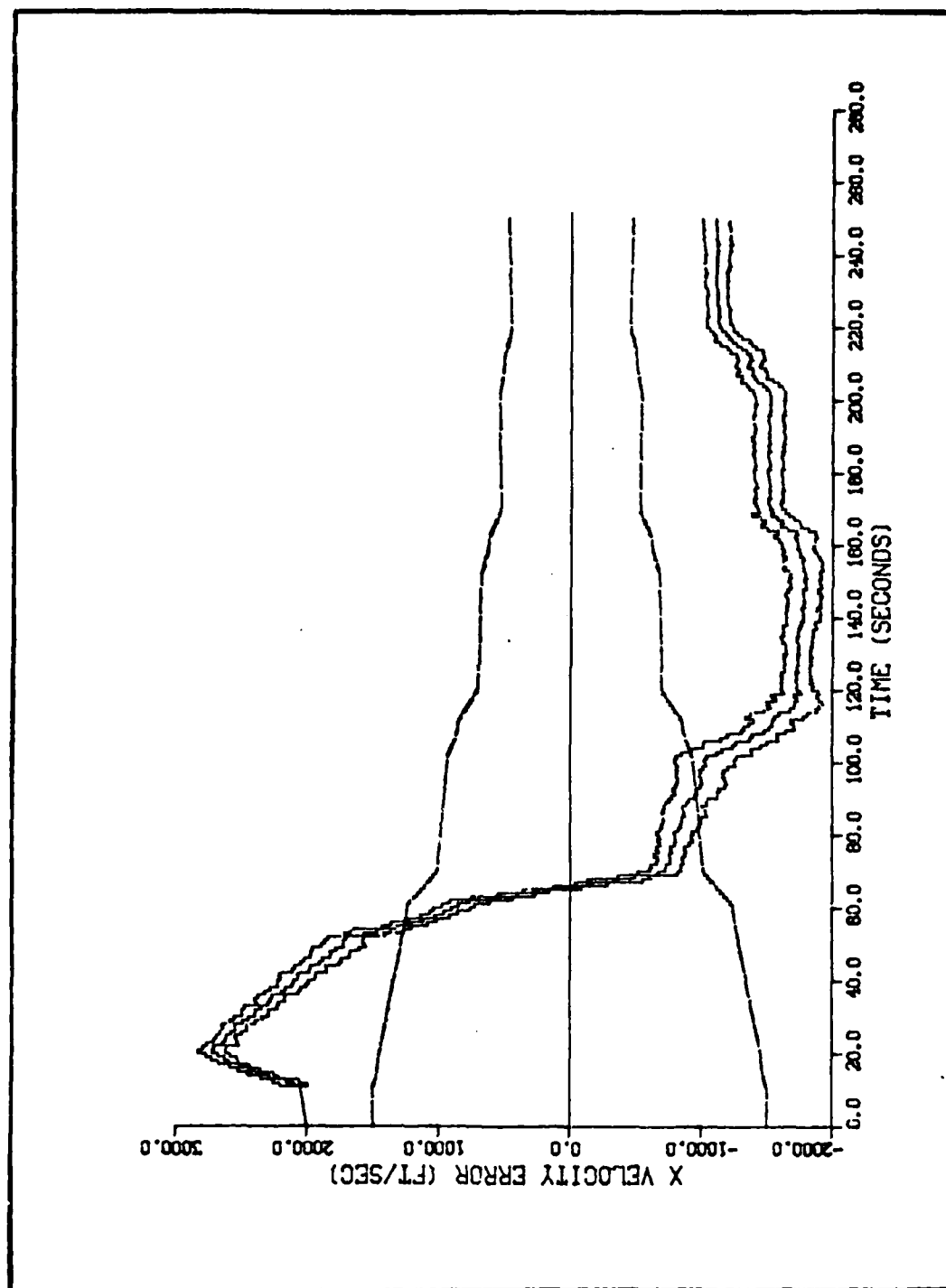


Figure M-28. Algorithm 6, Velocity Error in the X Direction Versus Time,
Target Profile 8

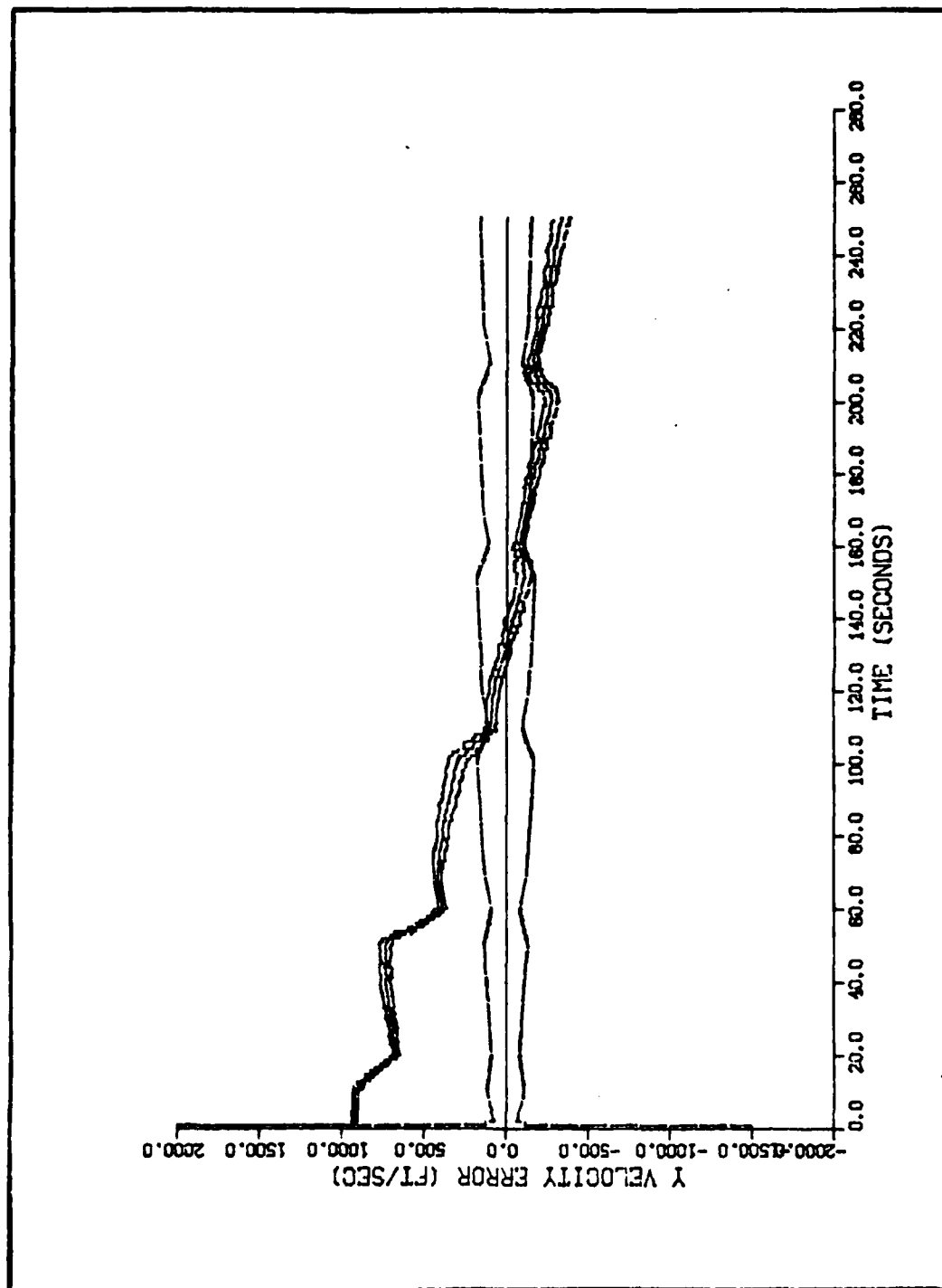


Figure M-29. Algorithm 6, Velocity Error in the Y Direction Versus Time, Target Profile 8

APPENDIX N
GRAPHICAL RESULTS OF ALGORITHM FIVE
AGAINST MANEUVERING TARGETS

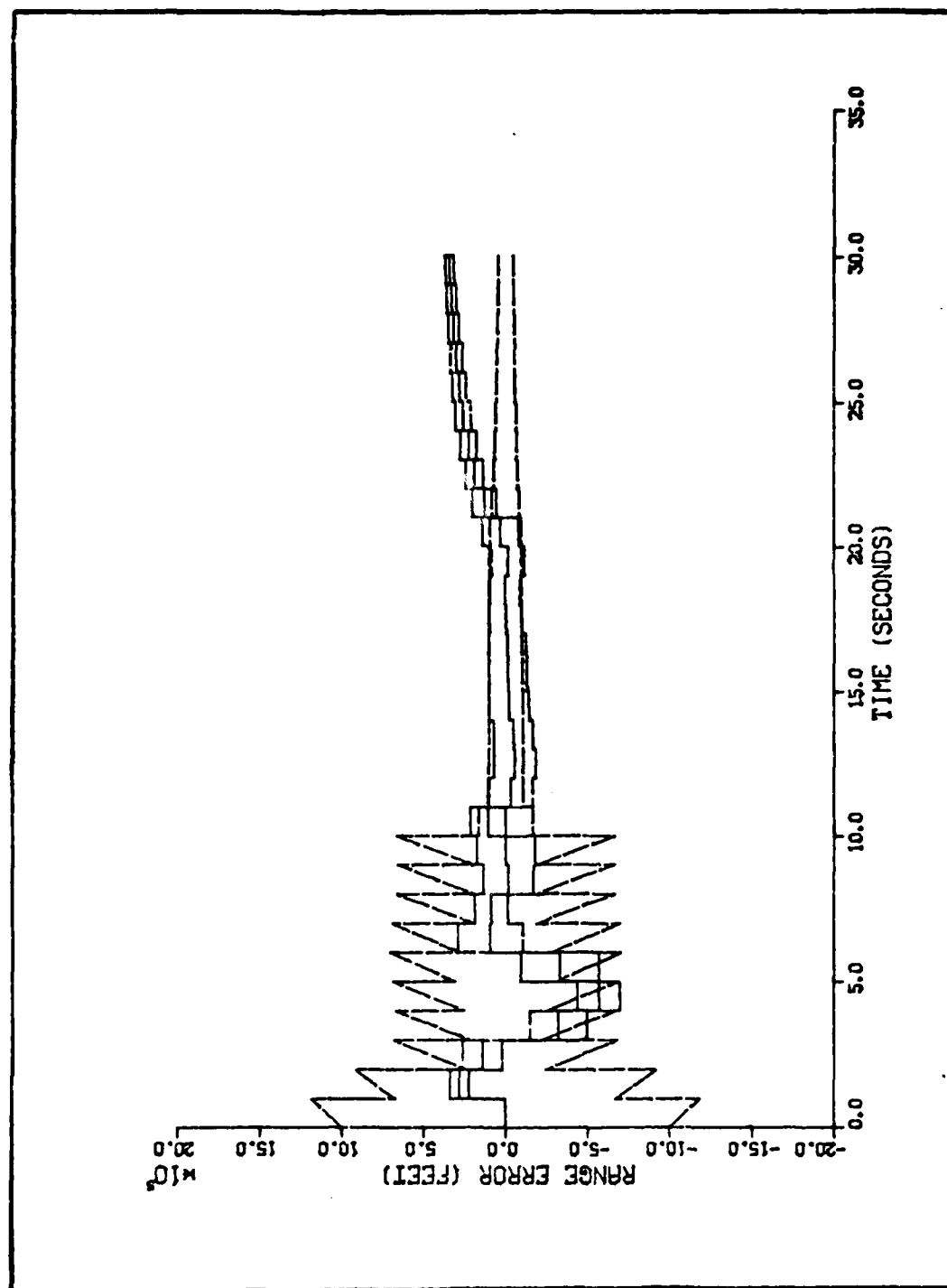


Figure N-1. Algorithm 5, Range Error Versus Time, Target Maintaining Collision Course

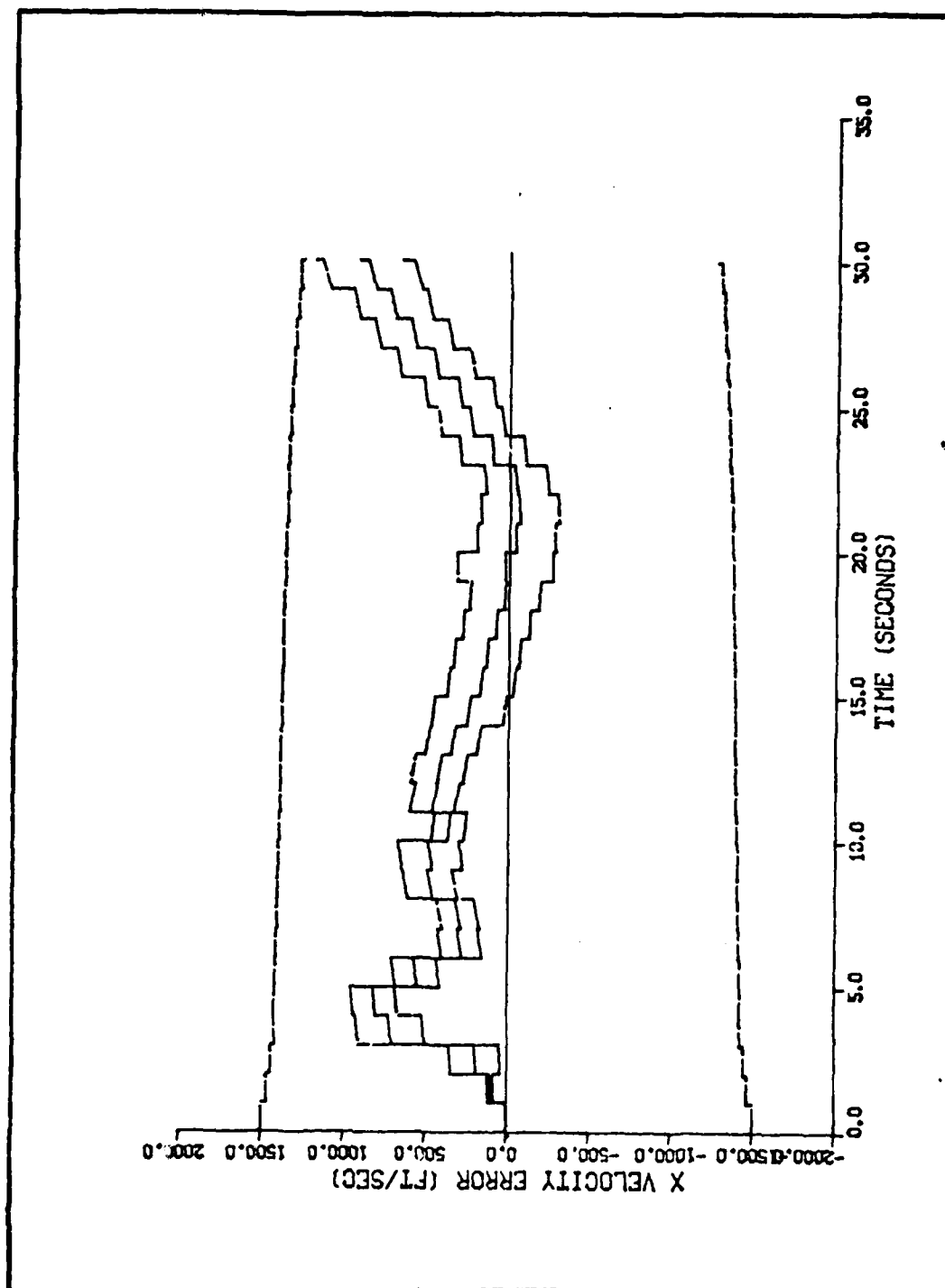


Figure N-2. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Maintaining Collision Course

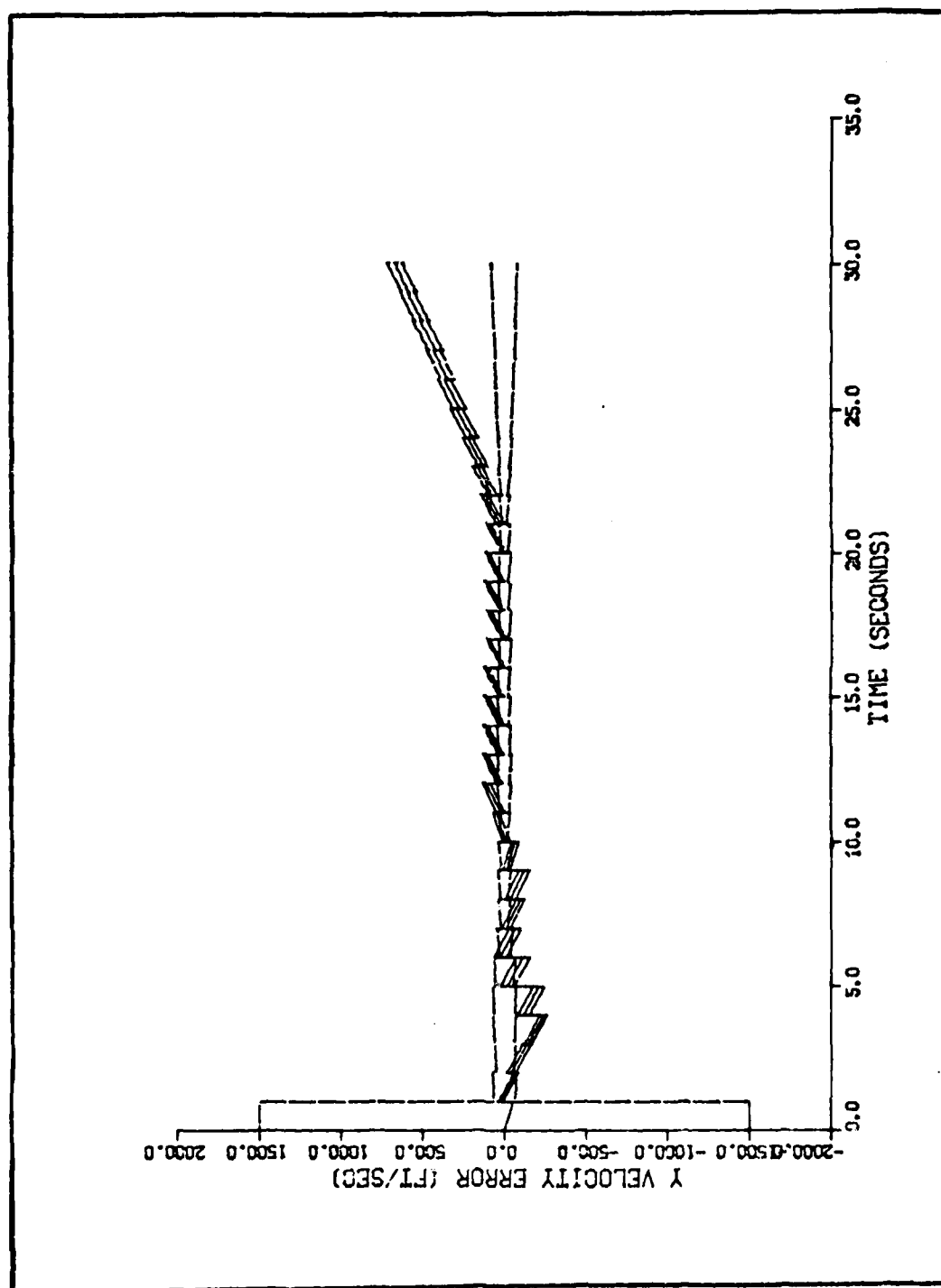


Figure N-3. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Maintaining Collision Course

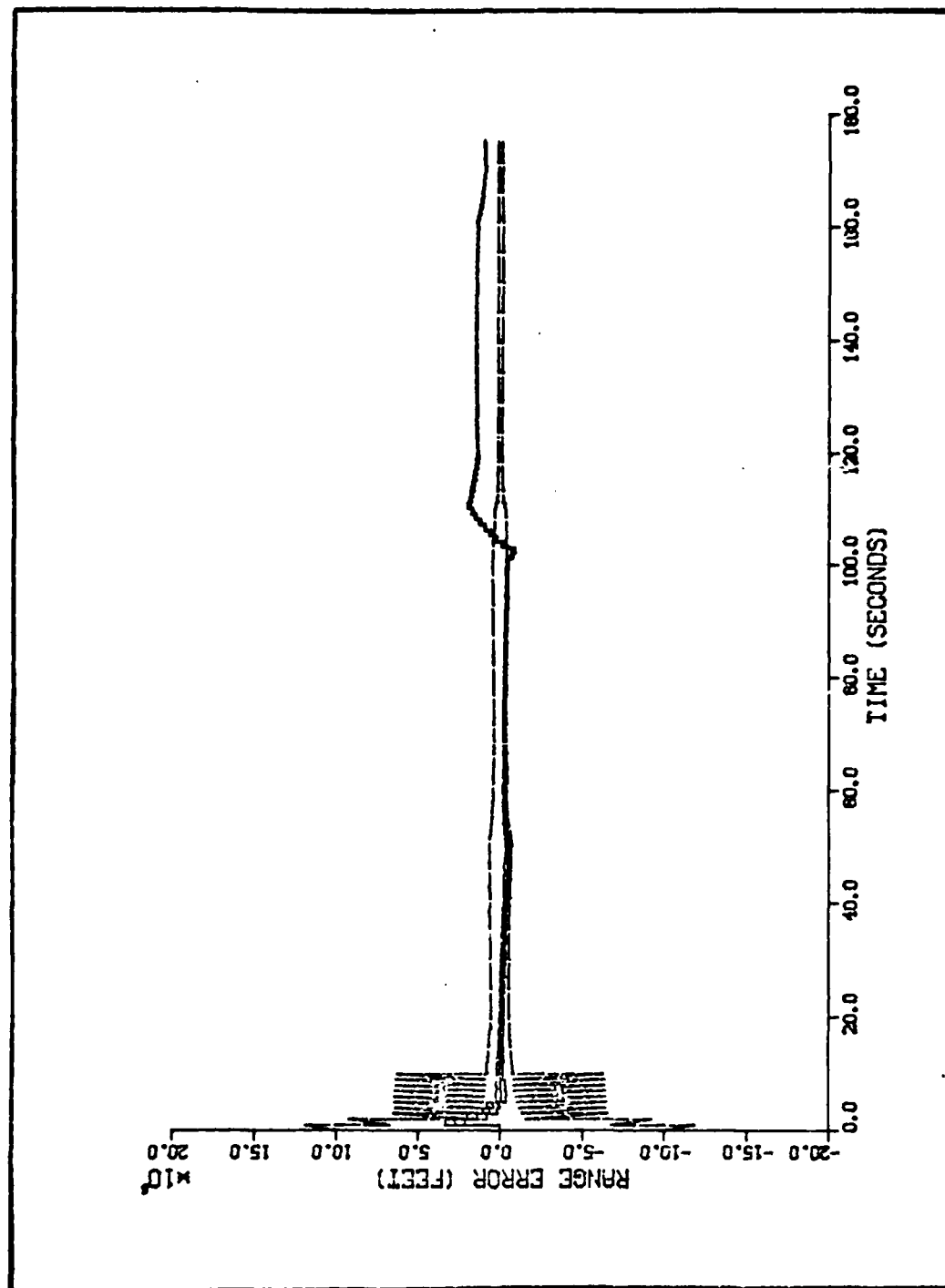


Figure N-4. Algorithm 5, Range Error Versus Time, Target Profile 4A

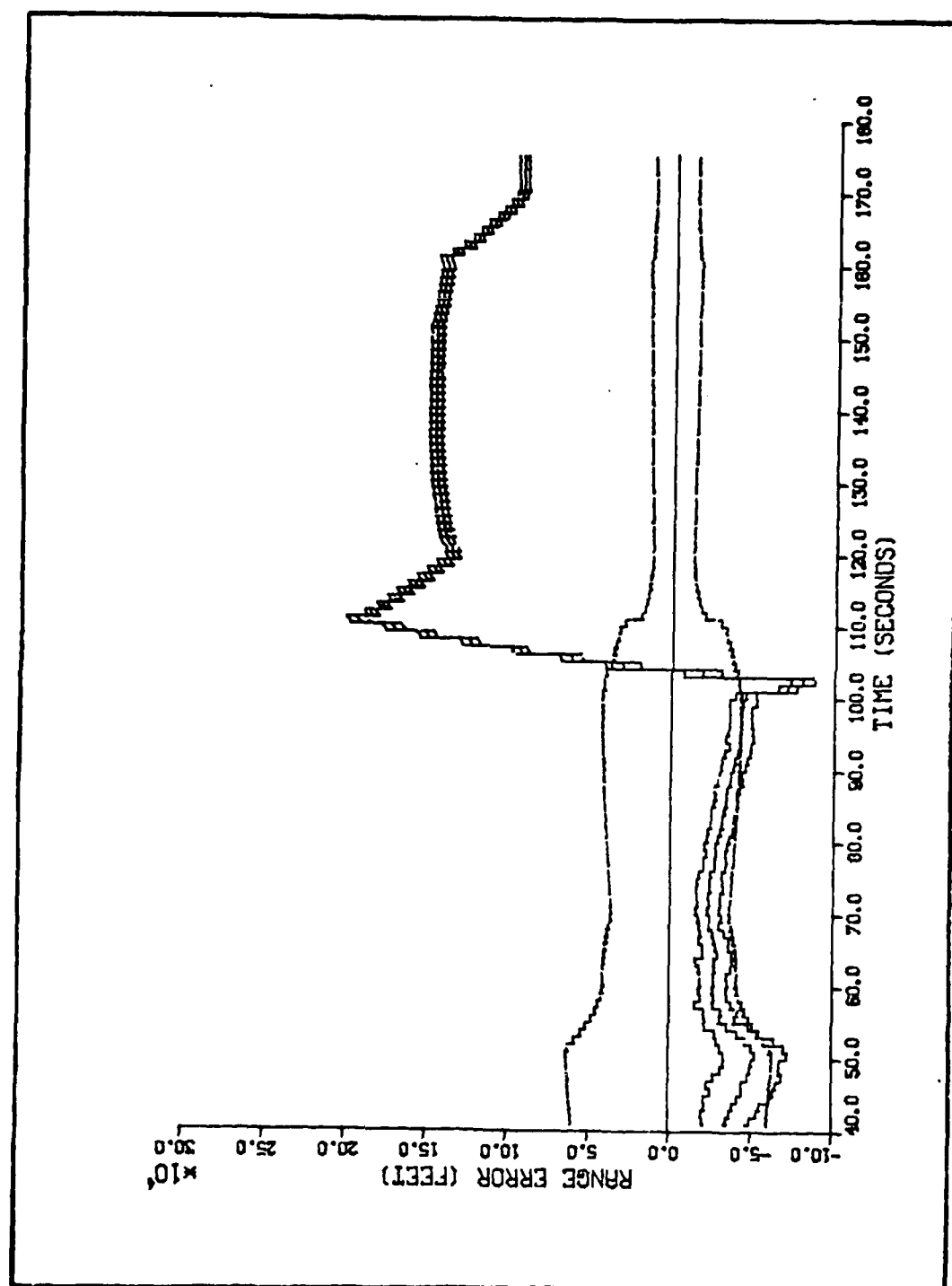


Figure N-5. Algorithm 5, Range Error Versus Time, Target Profile 4A

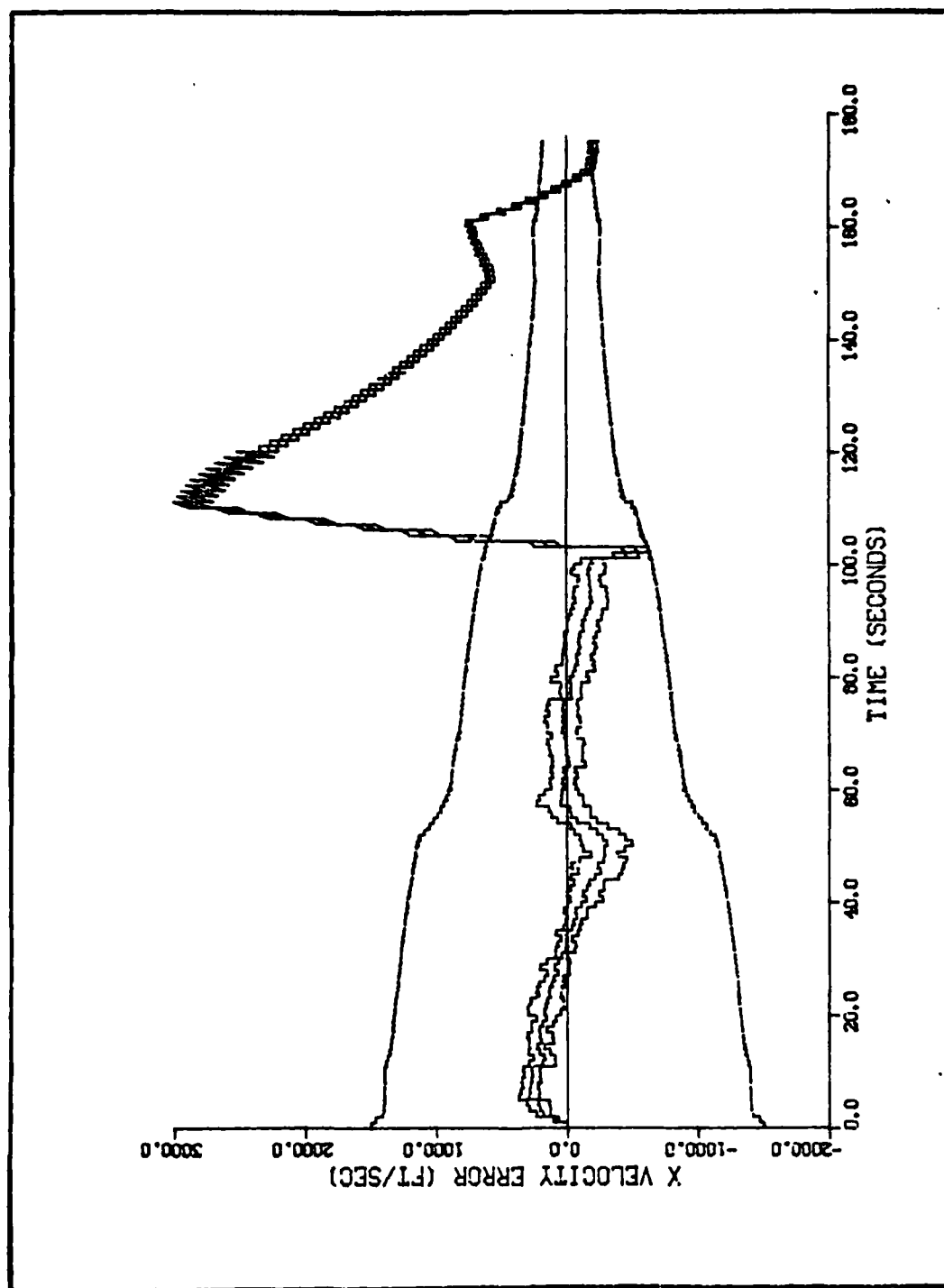


Figure N-6. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 4A

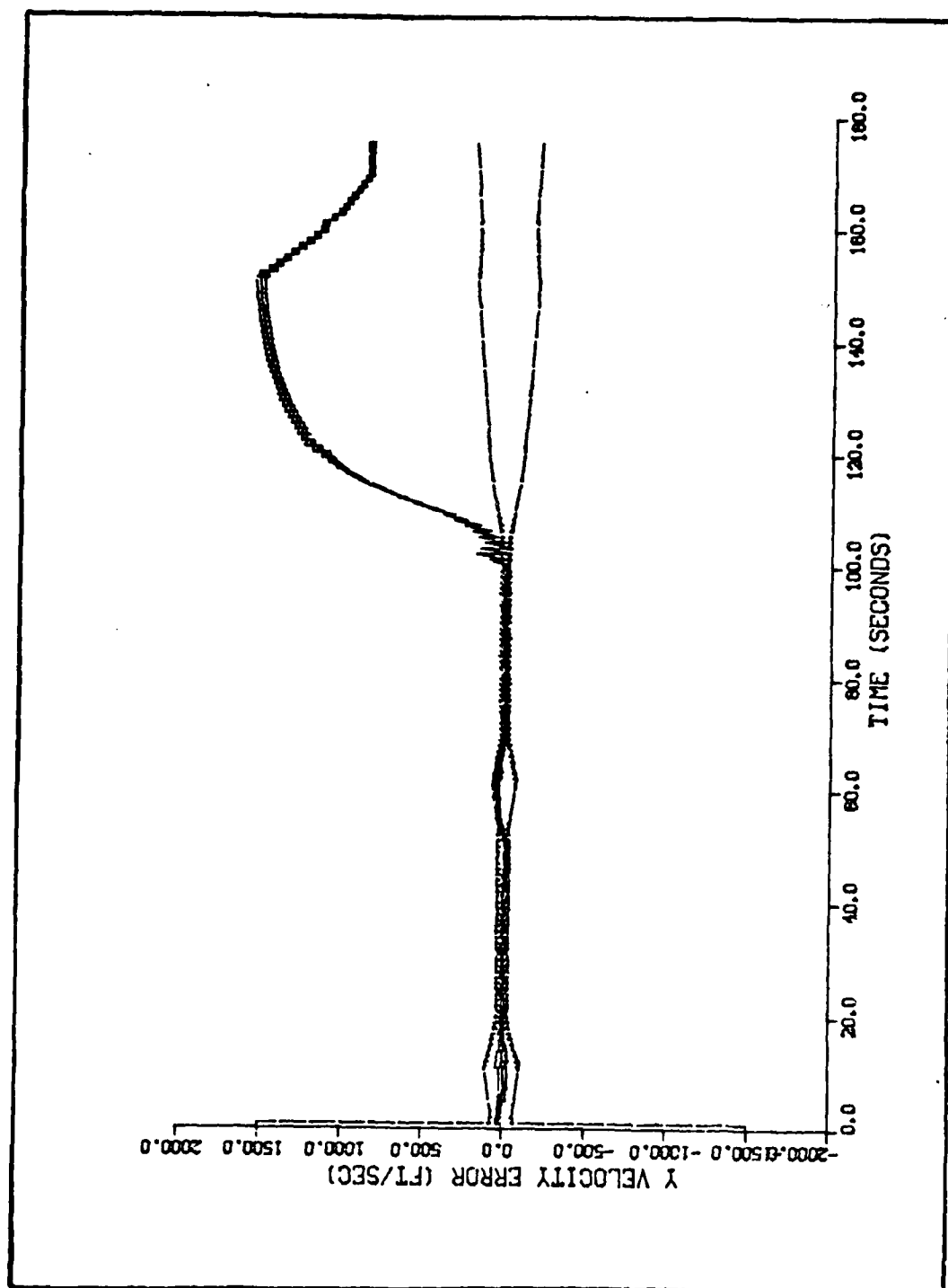


Figure N-7. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 4A

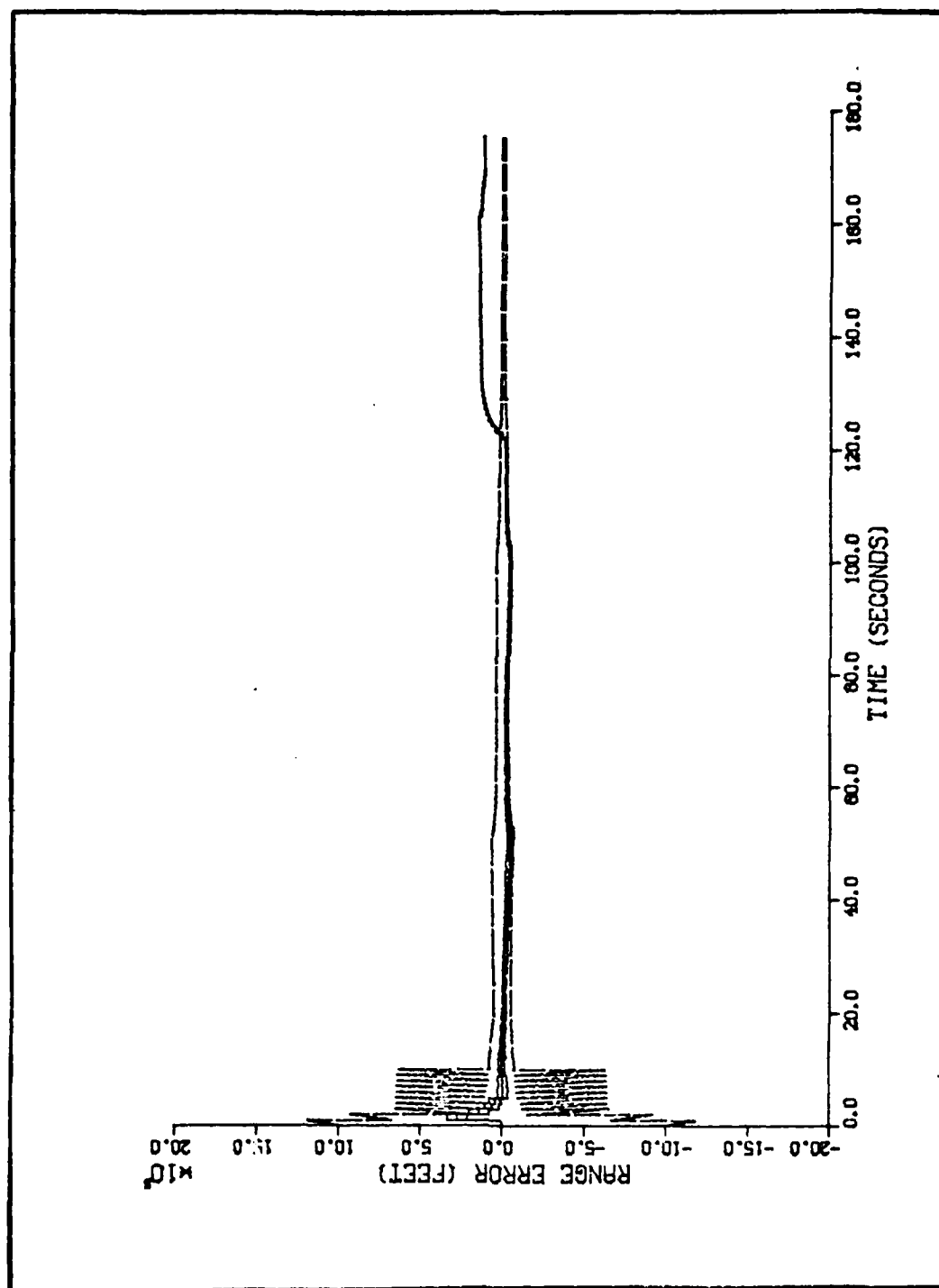


Figure N-8. Algorithm 5, Range Error Versus Time, Target Profile 4B

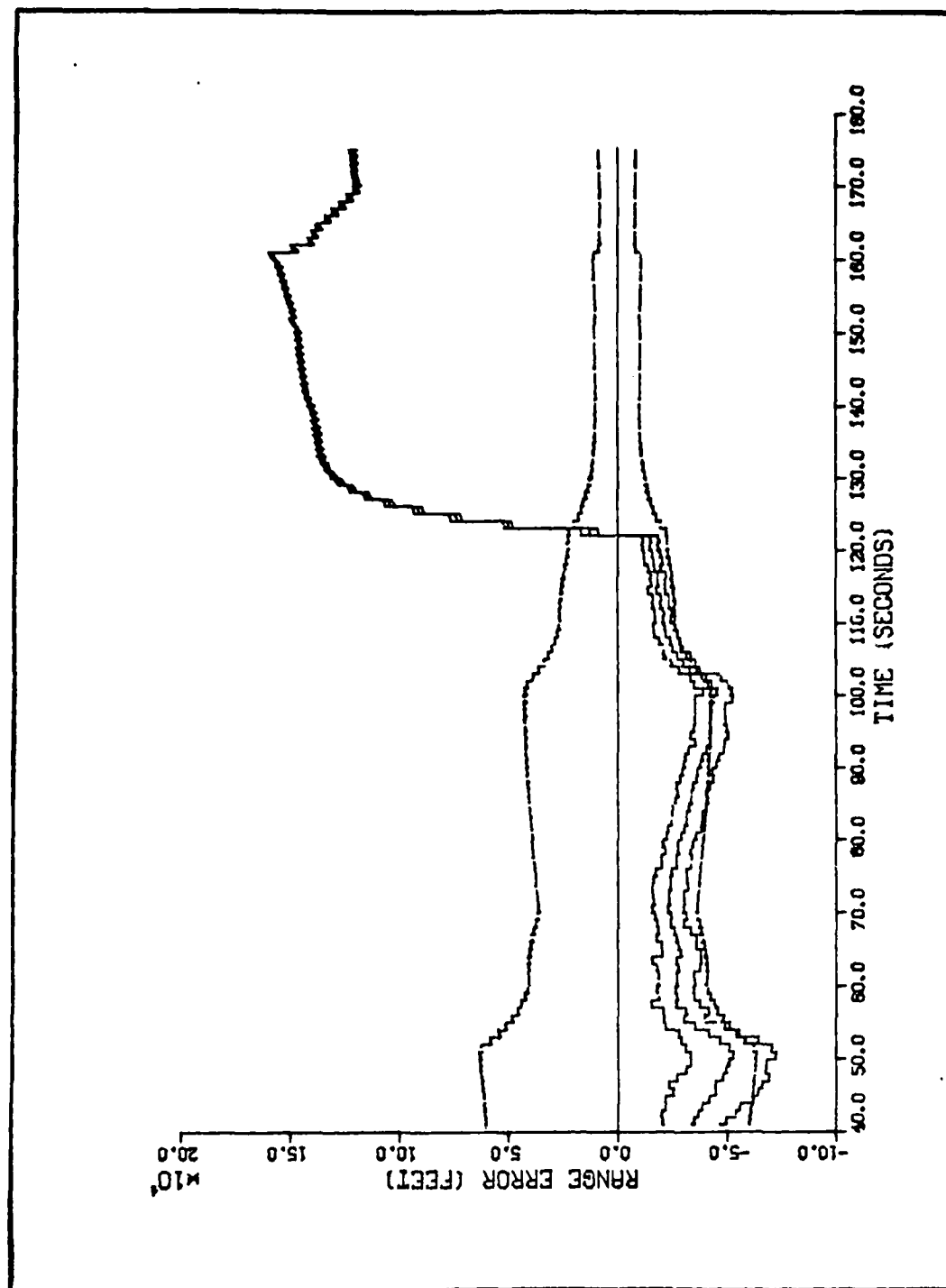


Figure N-9. Algorithm 5, Range Error Versus Time, Target Profile 4B

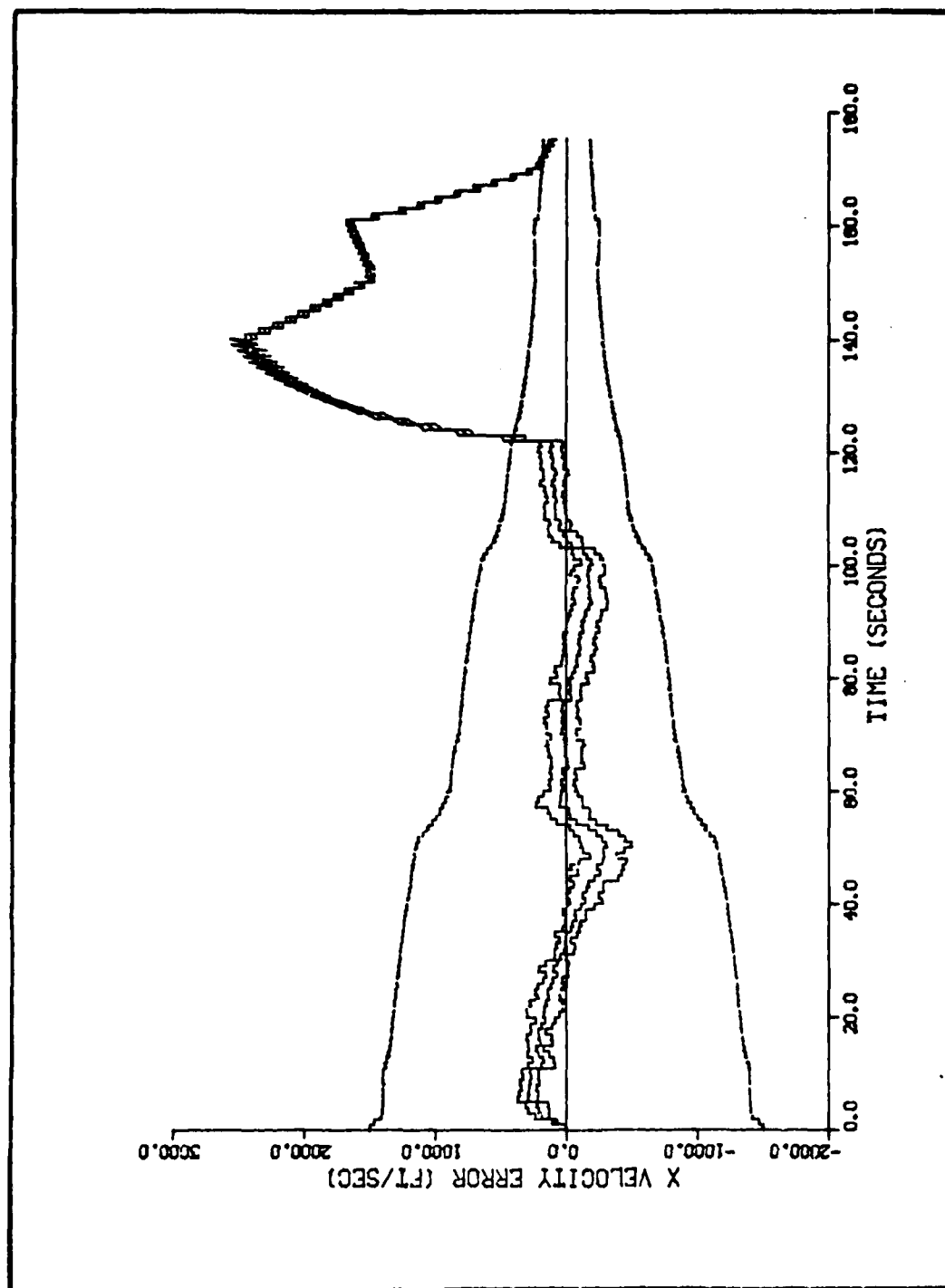


Figure N-10. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 4B

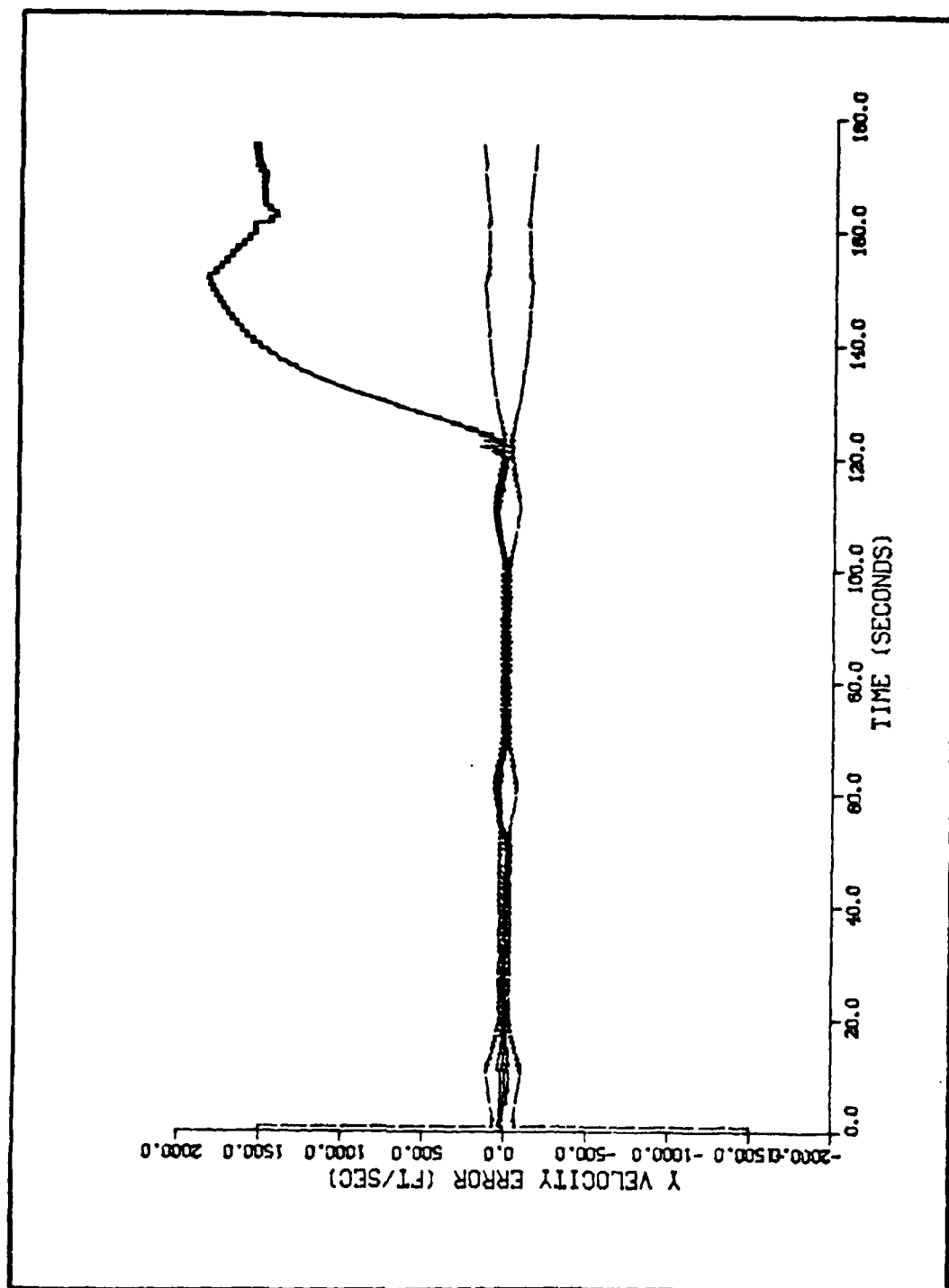


Figure N-11. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 4B

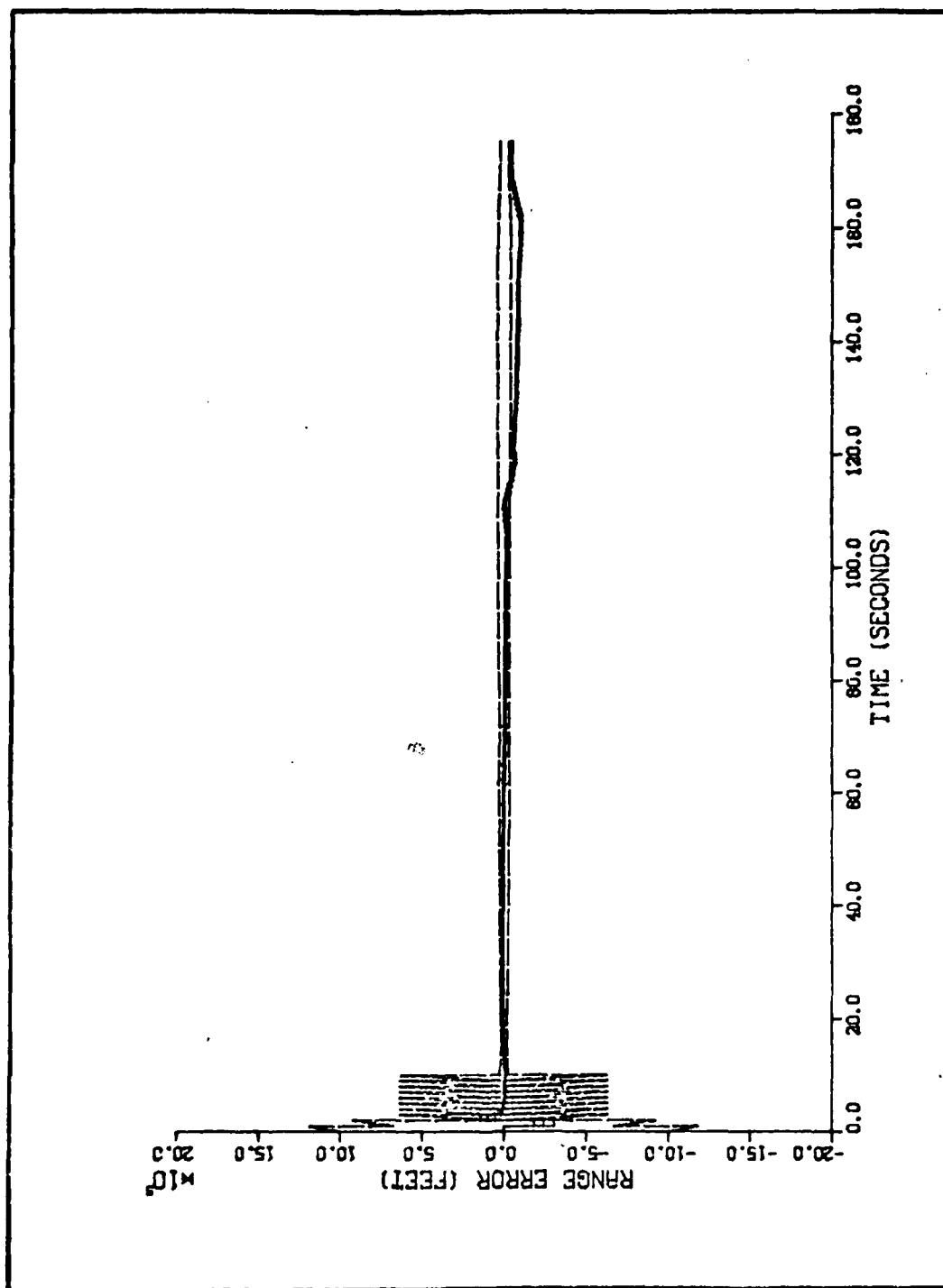


Figure N-12. Algorithm 5, Range Error Versus Time, Target Profile 5A

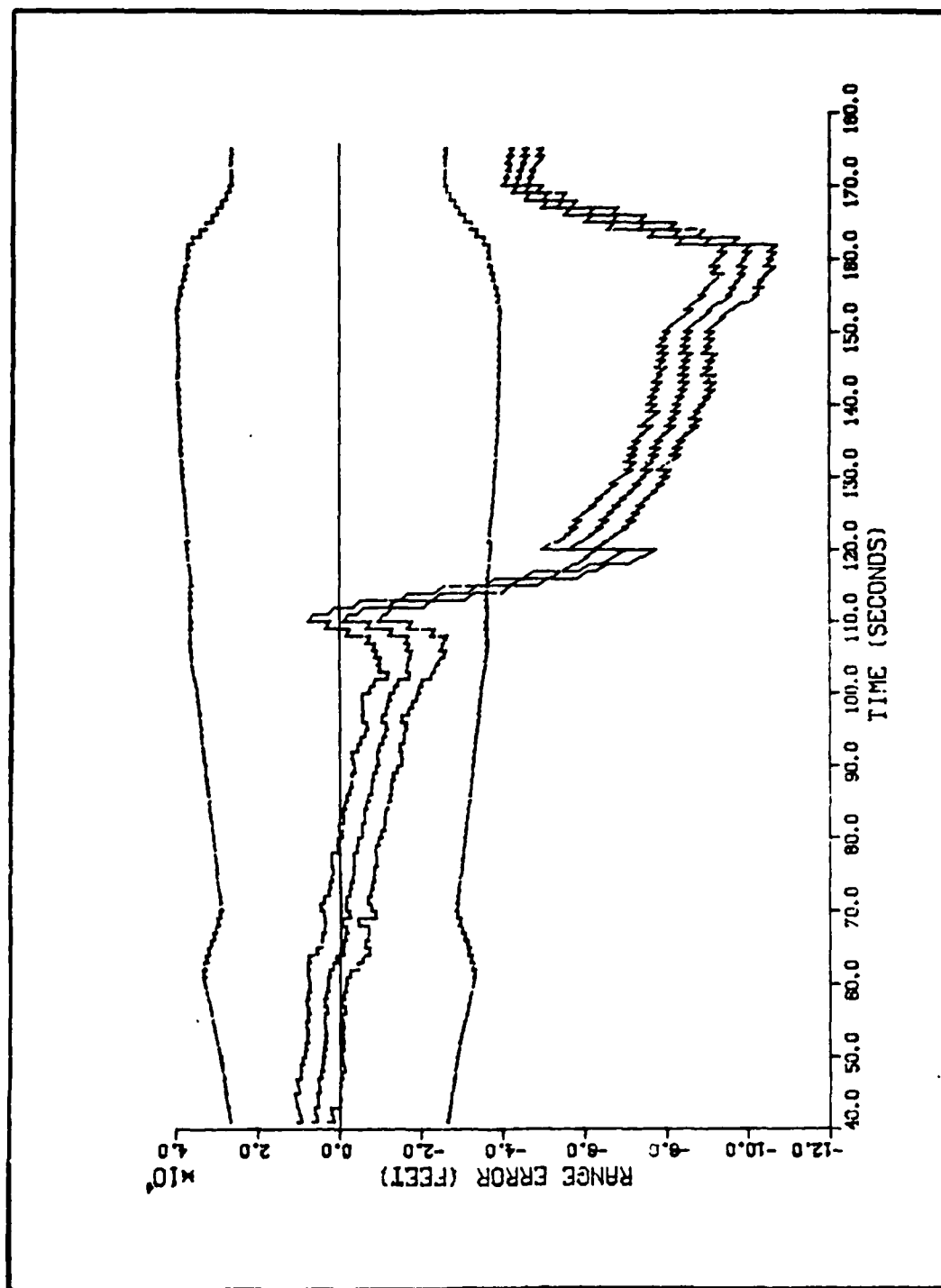


Figure N-13. Algorithm 5, Range Error Versus Time, Target Profile 5A

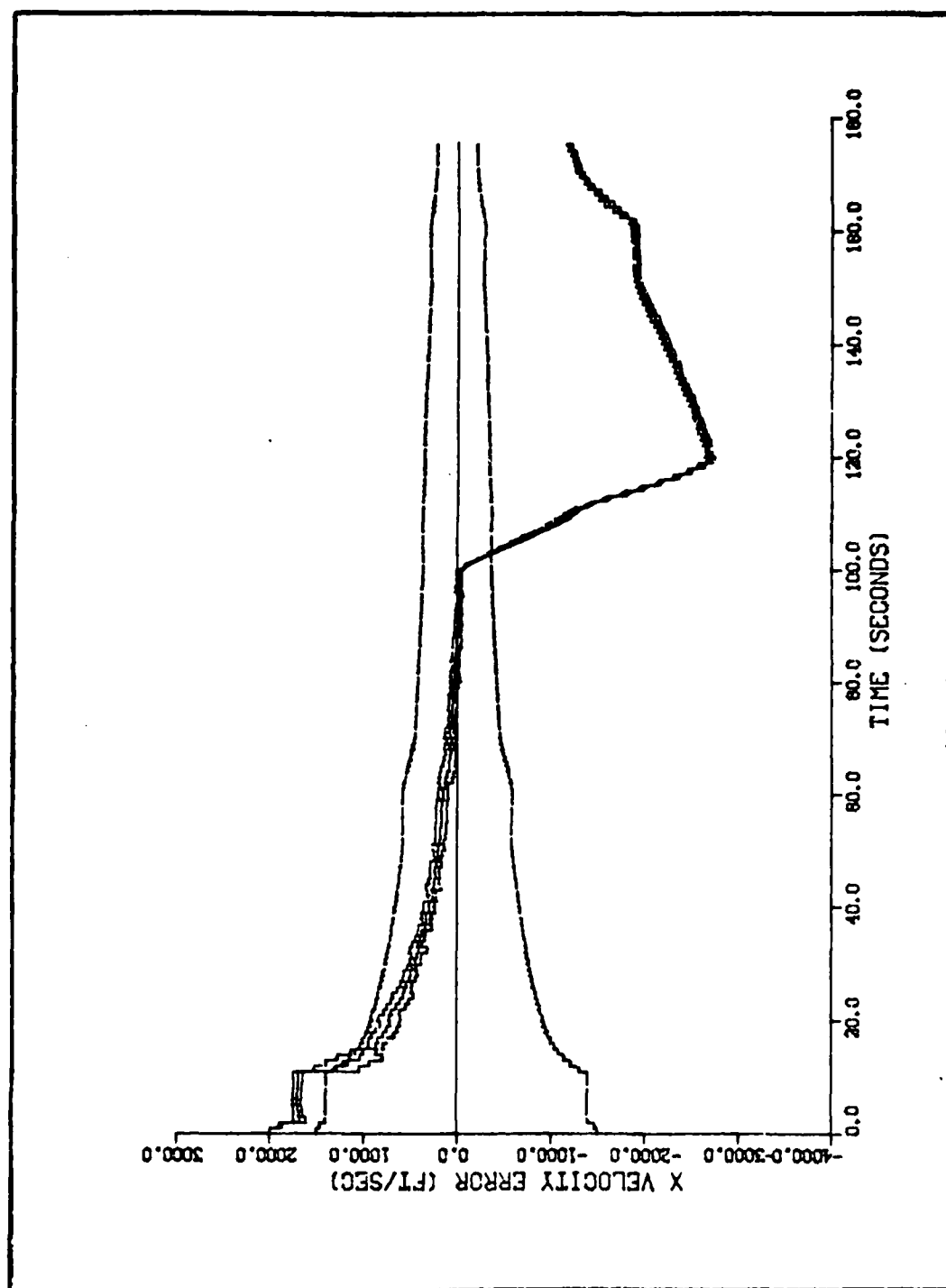


Figure N-14. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 5A

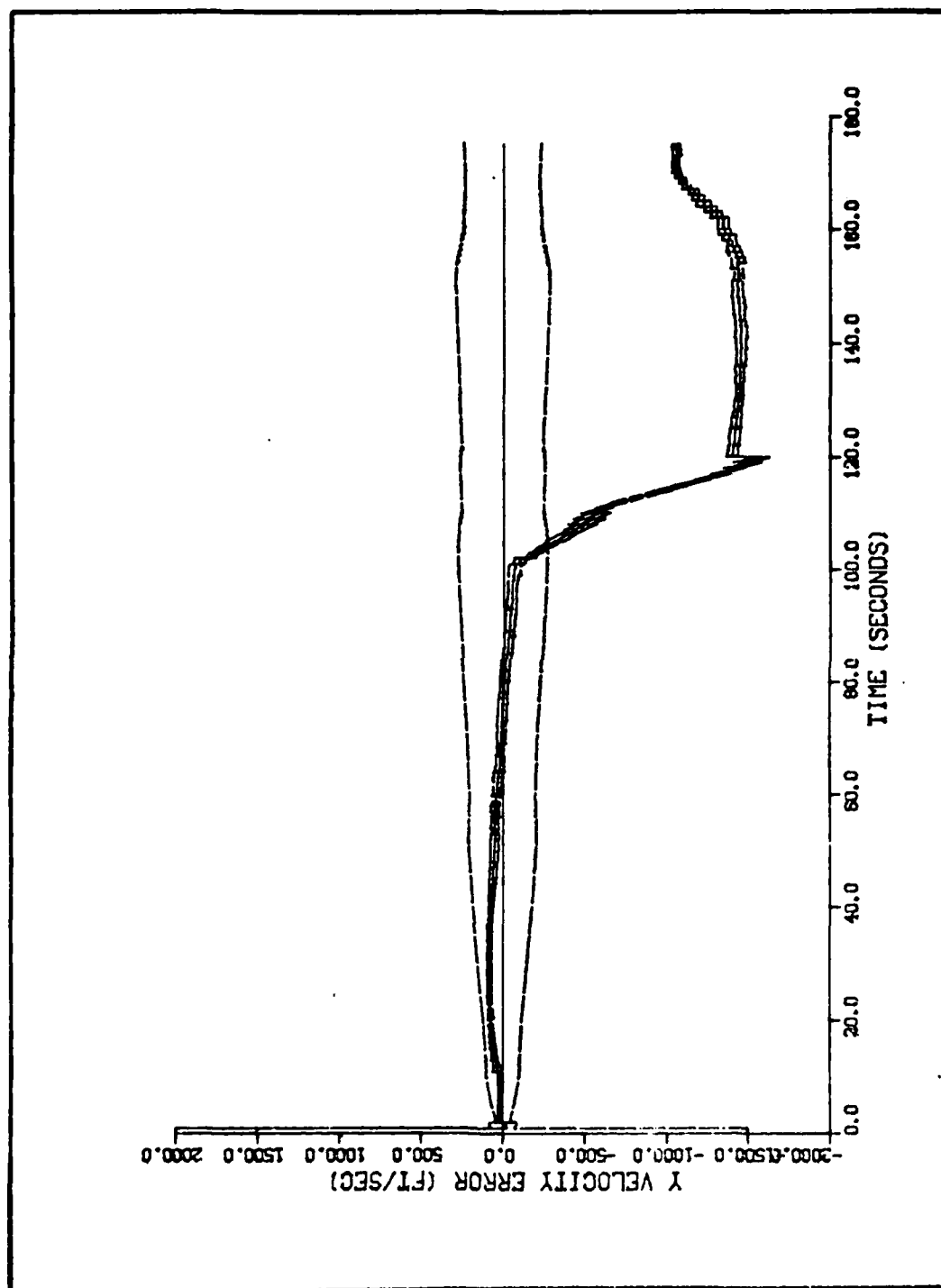


Figure N-15. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 5A

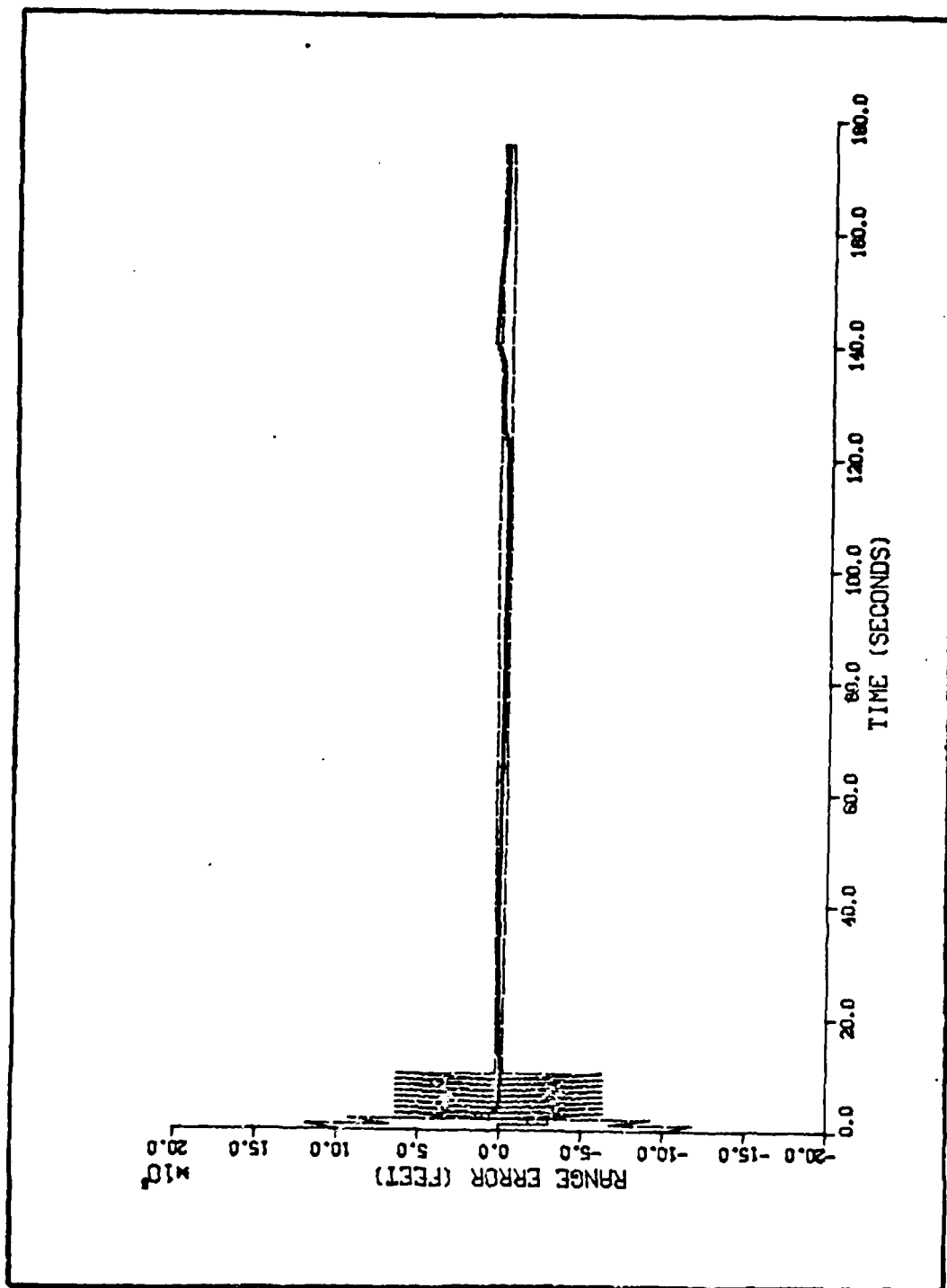


Figure N-16. Algorithm 5, Range Error Versus Time, Target Profile 5B

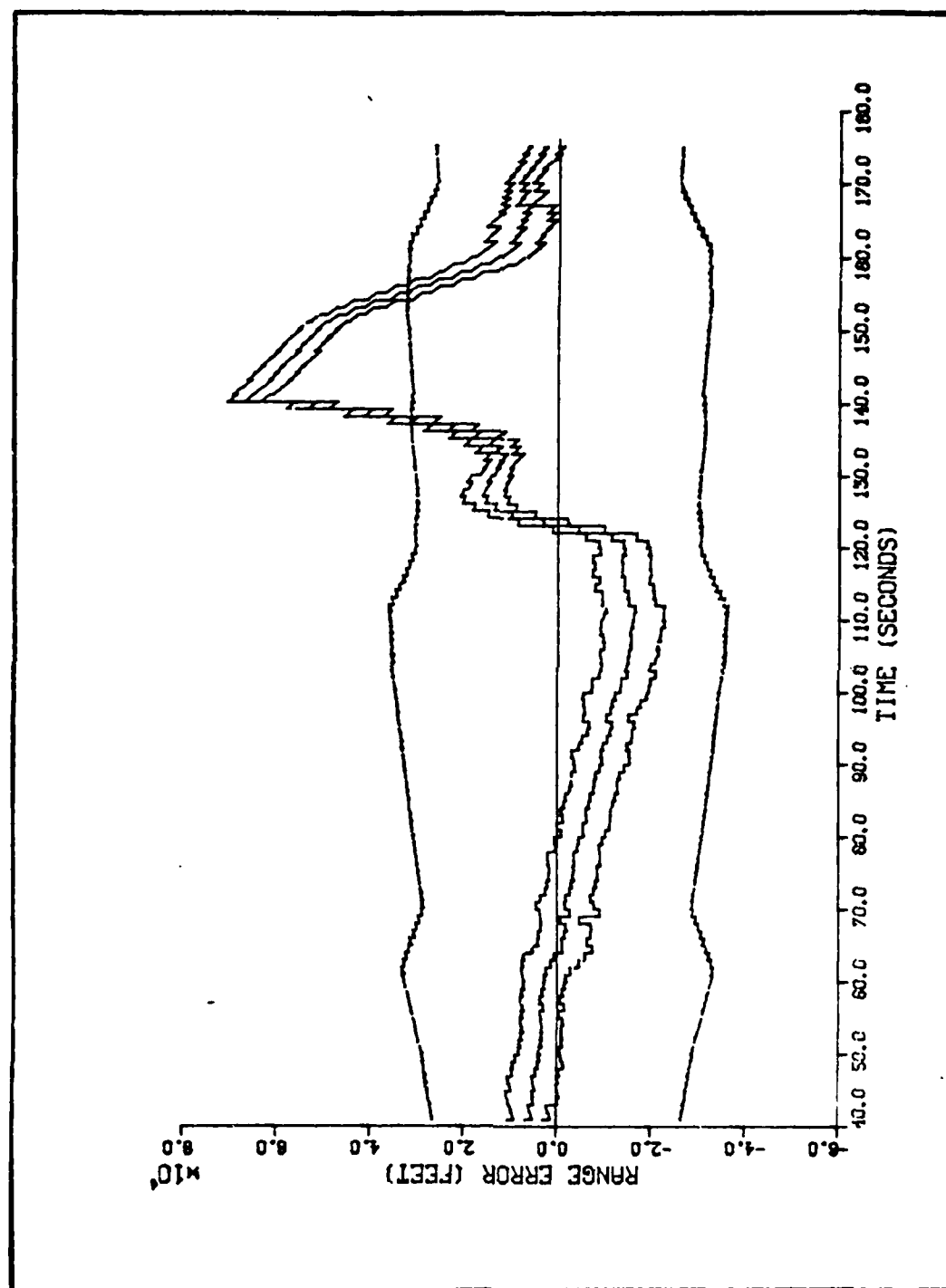


Figure N-17. Algorithm 5, Range Error Versus Time, Target Profile 5B

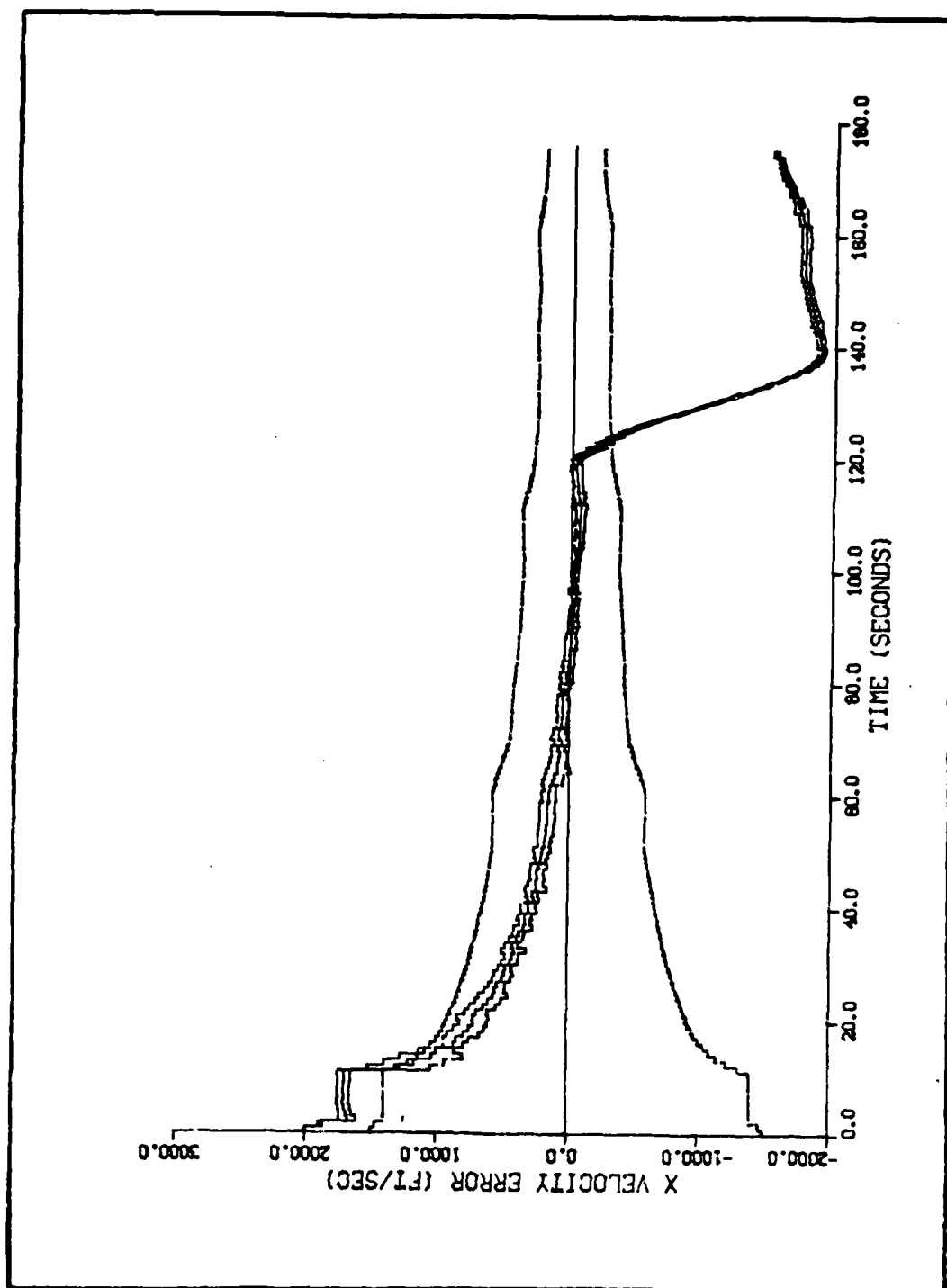


Figure N-18. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 5B

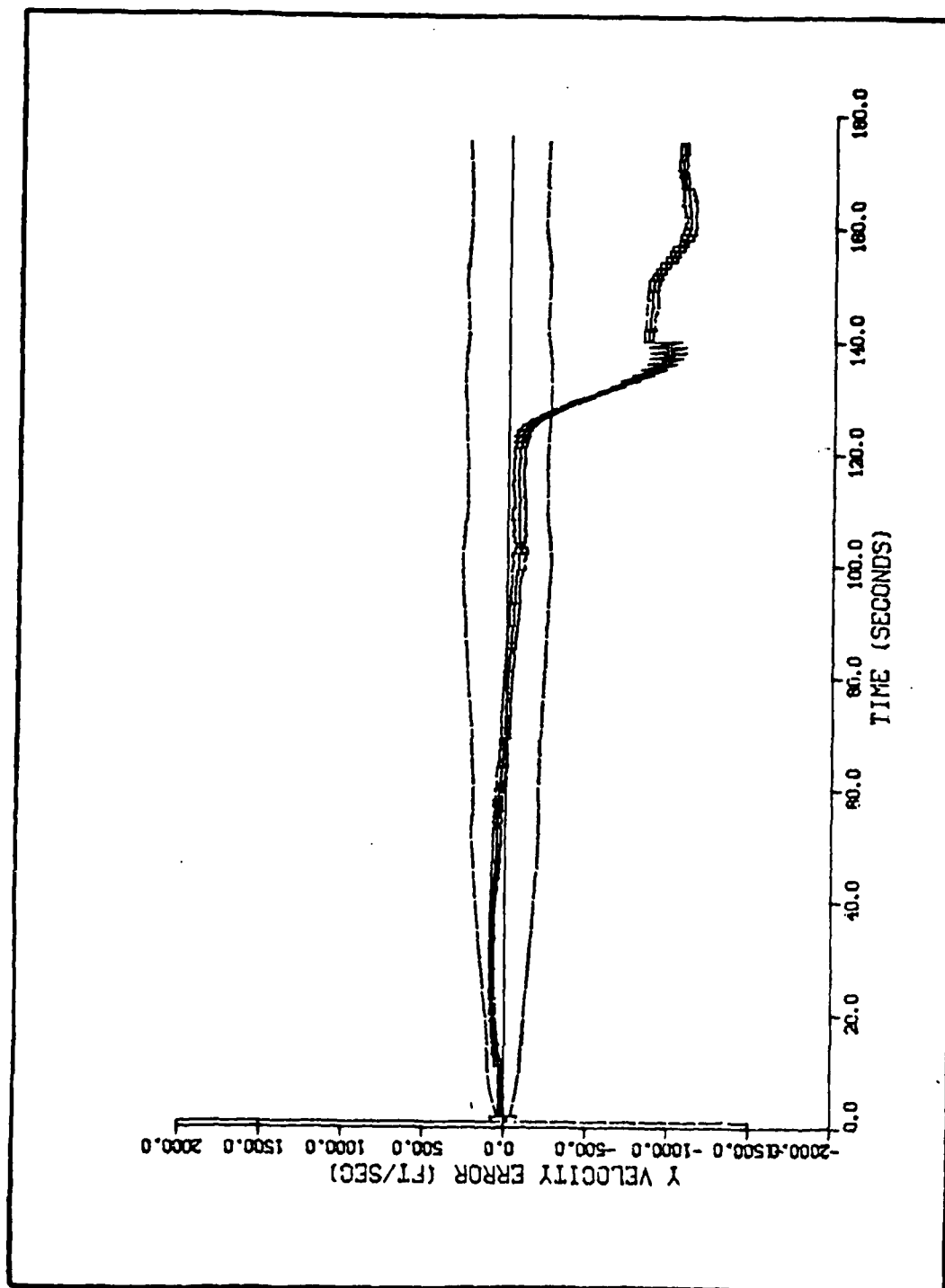


Figure N-19. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 5B

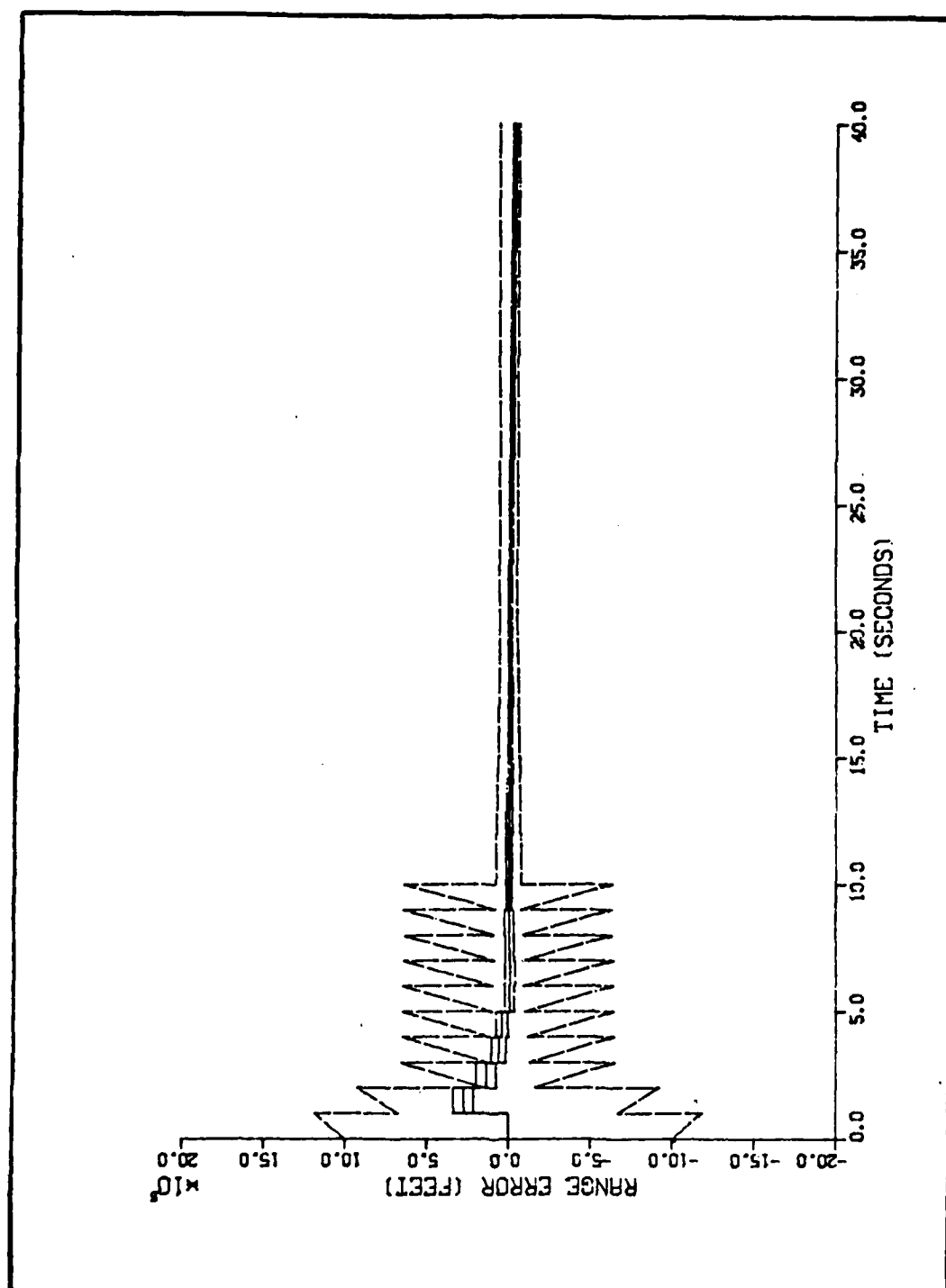


Figure N-2C. Algorithm 5, Range Error Versus Time, Target Profile 4C

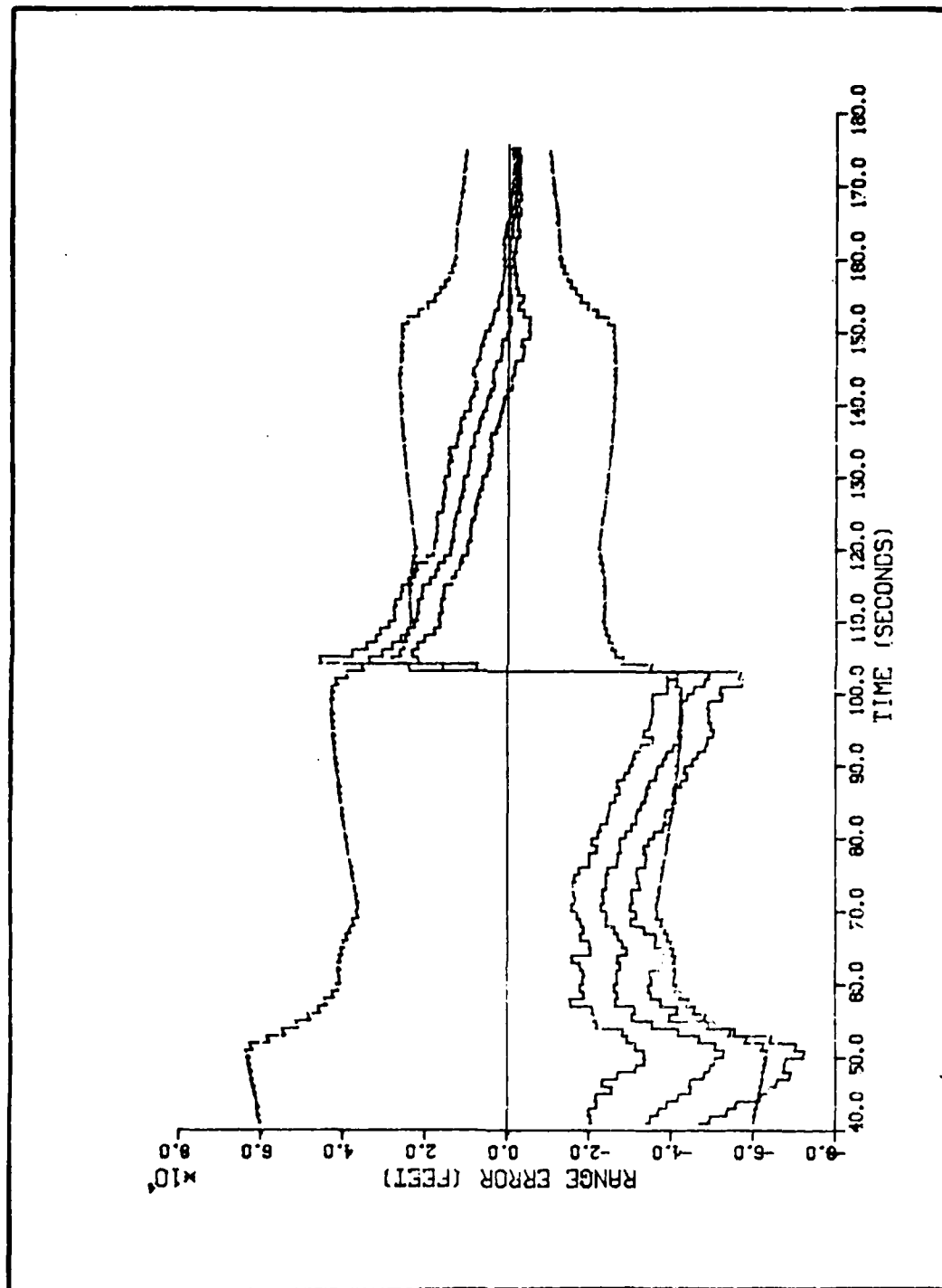


Figure N-21. Algorithm 5, Range Error Versus Time, Target Profile 4C

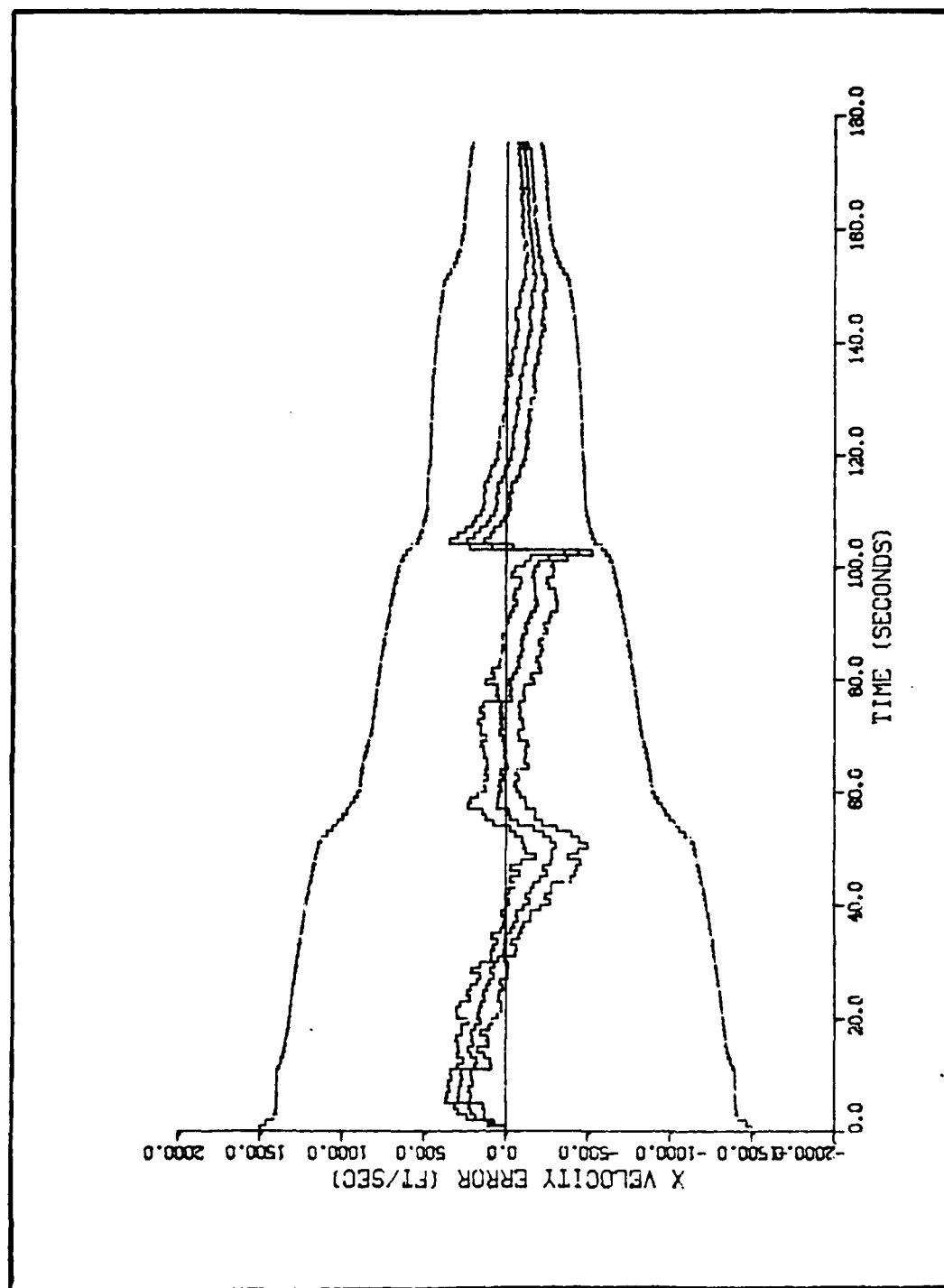


Figure N-22. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 4C

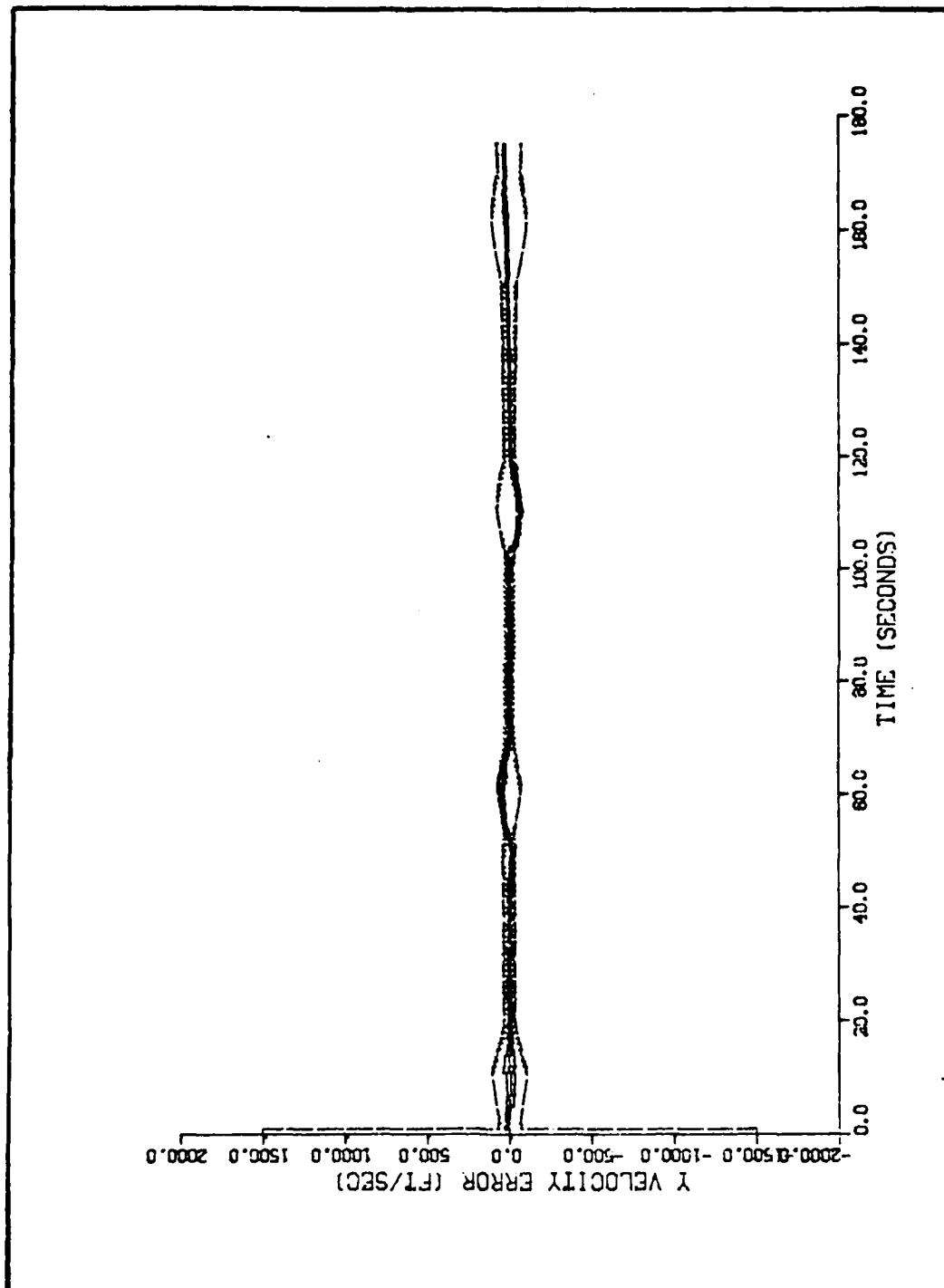


Figure N-23. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 4C

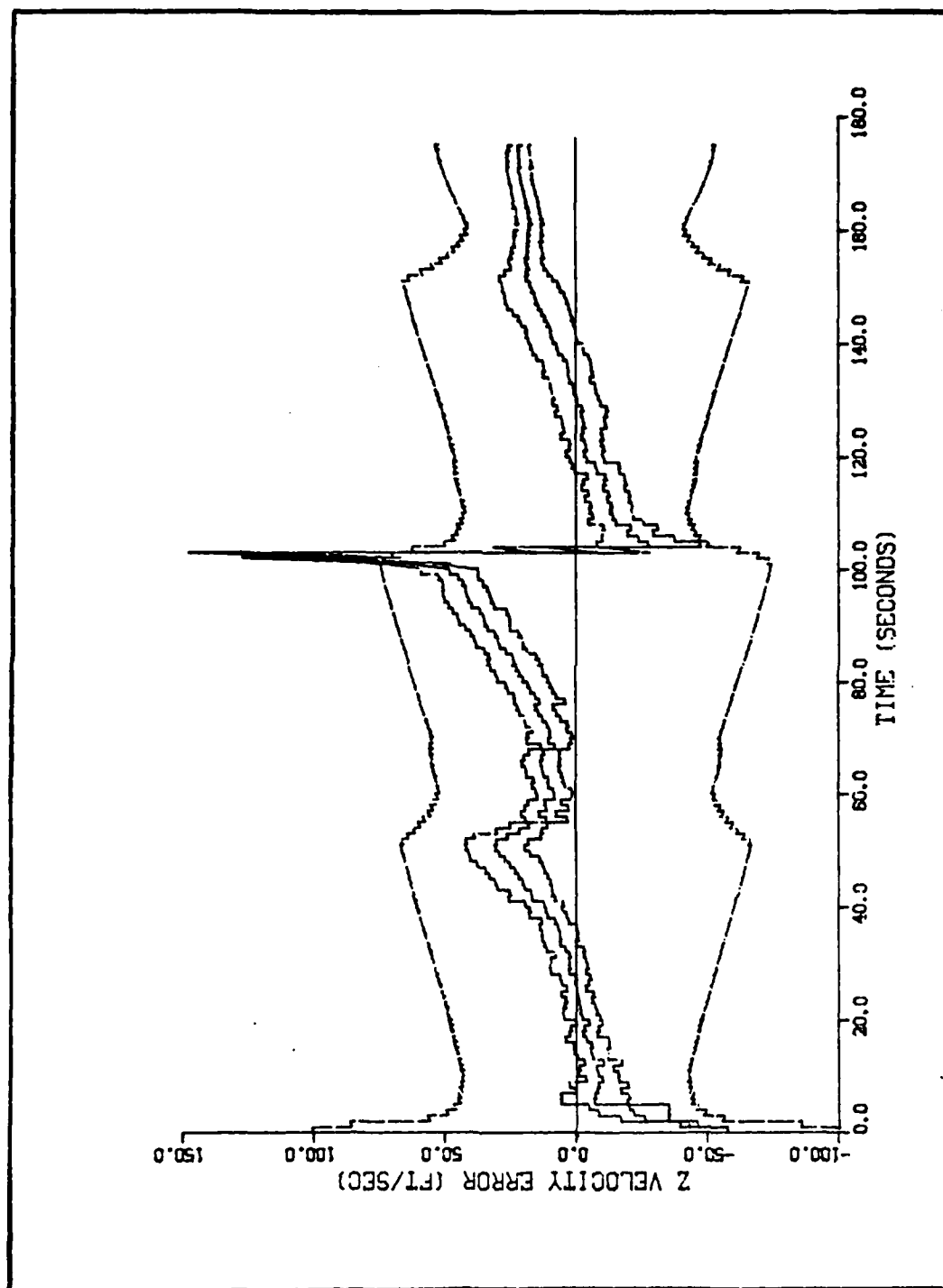


Figure N-24. Algorithm 5, Velocity Error in the Z Direction Versus Time, Target Profile 4C

APPENDIX O

GRAPHICAL RESULTS OF ALGORITHM FIVE AGAINST
NONMANEUVERING TARGETS WITH VARYING
MEASUREMENT ERROR VARIANCES

Throughout this appendix, the standard deviation of the angle measurement errors is referred to as σ_{ANGLE} , and the standard deviation of the angle rate measurement errors as σ_{RATE} . When either σ_{ANGLE} or σ_{RATE} is not explicitly given in the plot caption, it is assumed to be the nominal of 100 μrad or 20 $\mu\text{rad/sec}$, respectively.

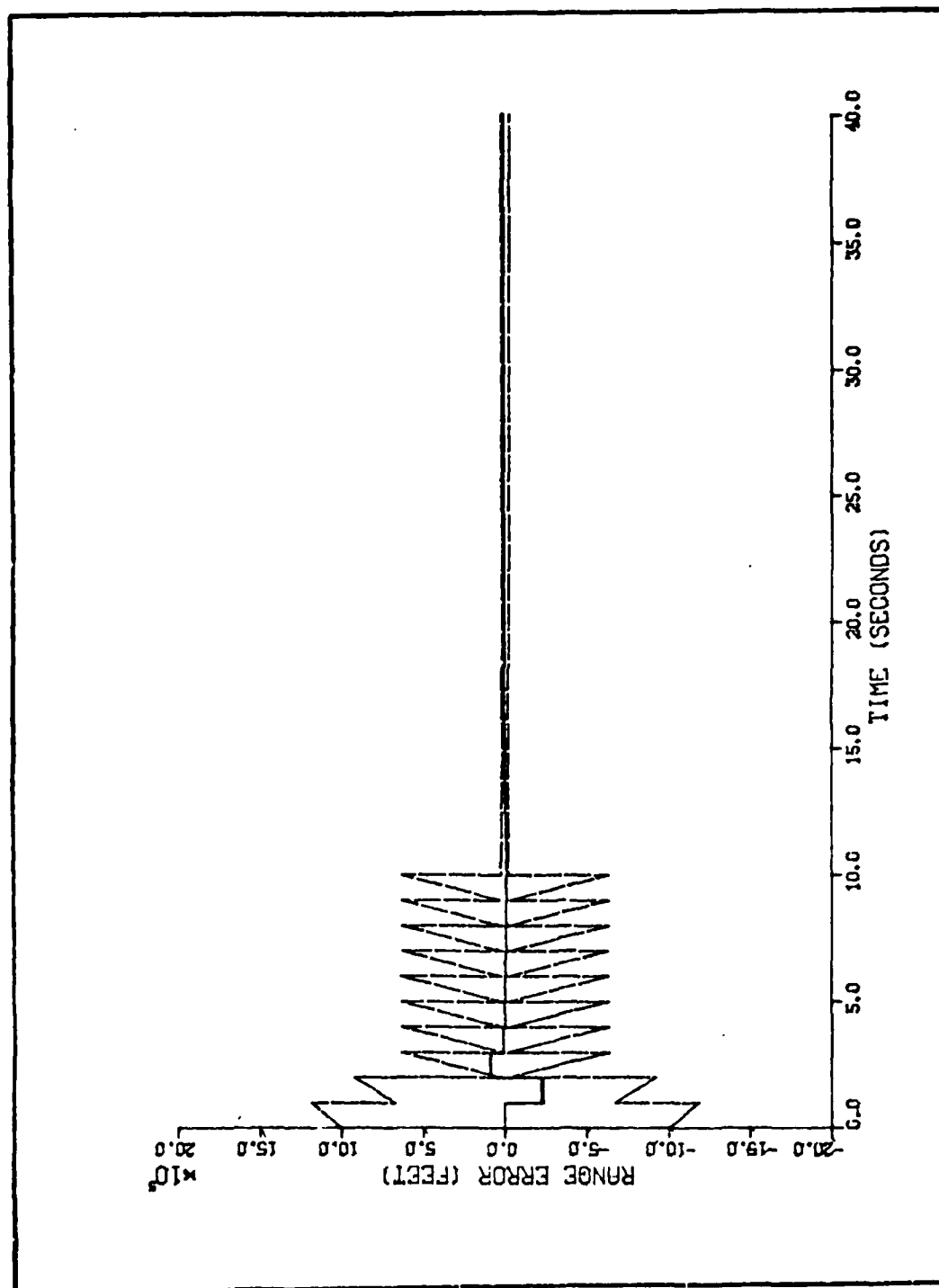


Figure O-1. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\sigma_{\text{RATE}} = 2 \text{ } \mu\text{rad/sec}$

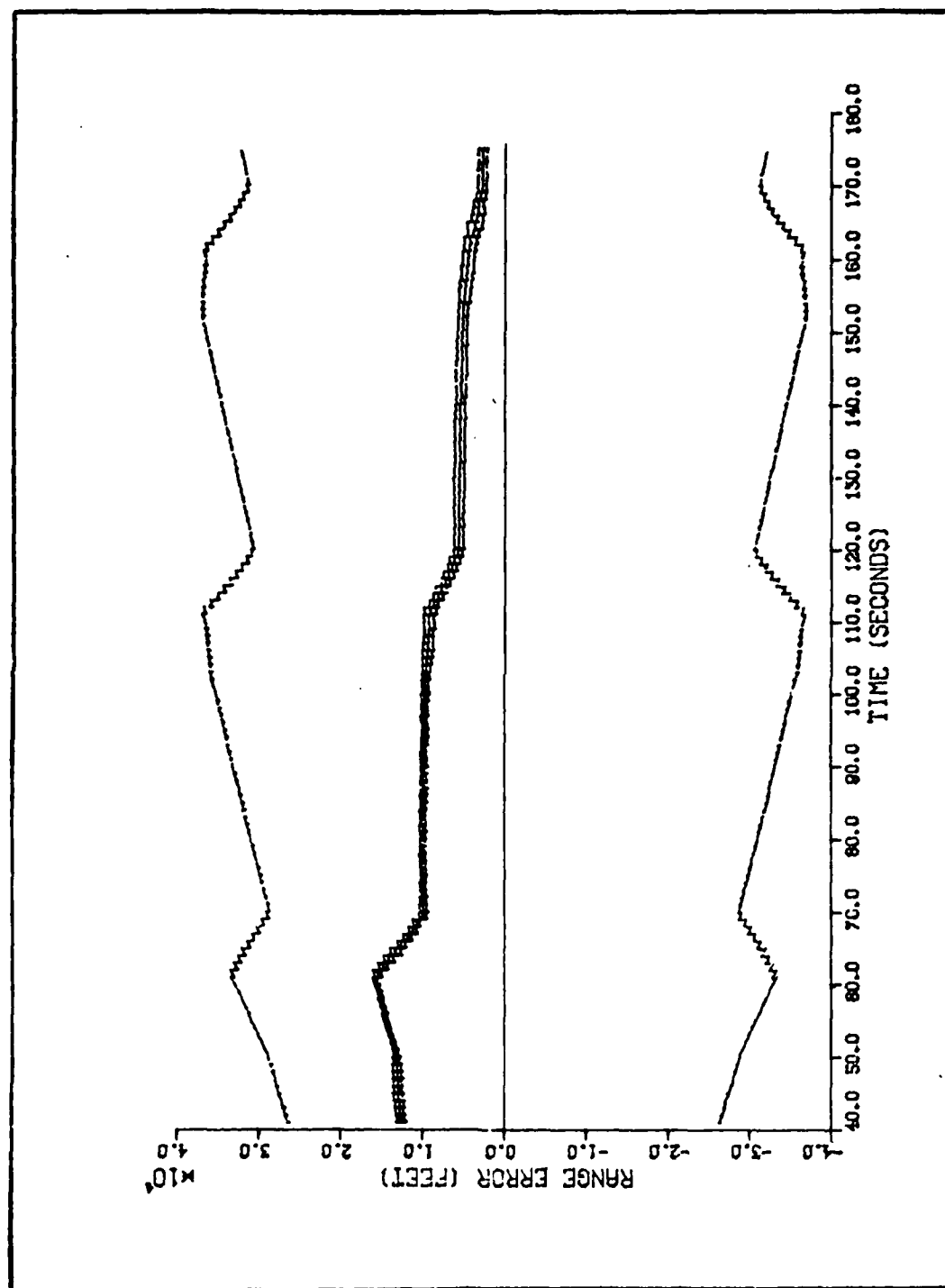


Figure 0-2. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\sigma_{\text{RATE}} = 2 \text{ } \mu\text{rad/sec}$

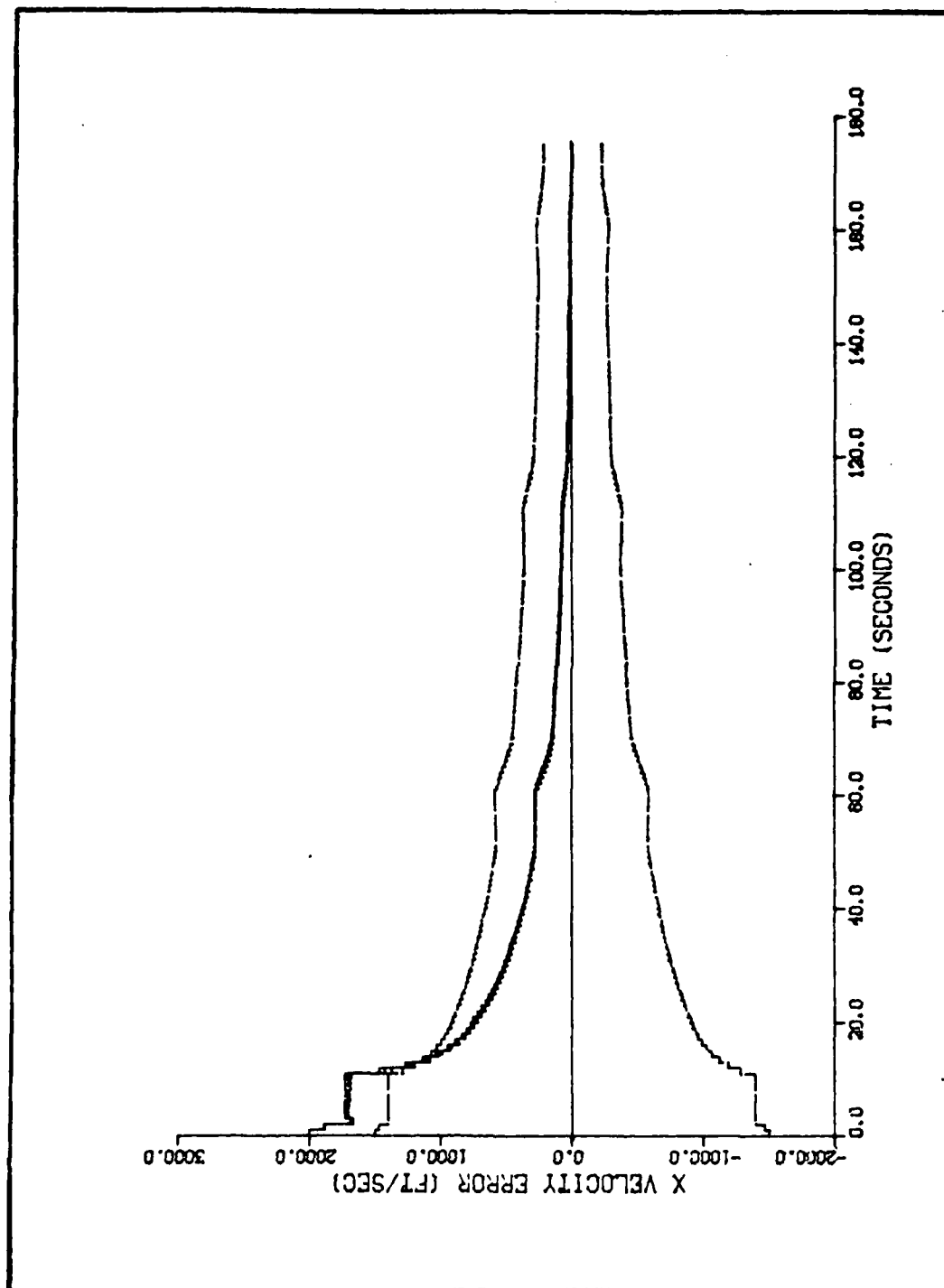


Figure 0-3. Algorithm 5, Velocity Error in the X Direction Versus Time,
Target Profile 5, $\sigma_{\text{RATE}} = 2 \text{ } \mu\text{rad/sec}$

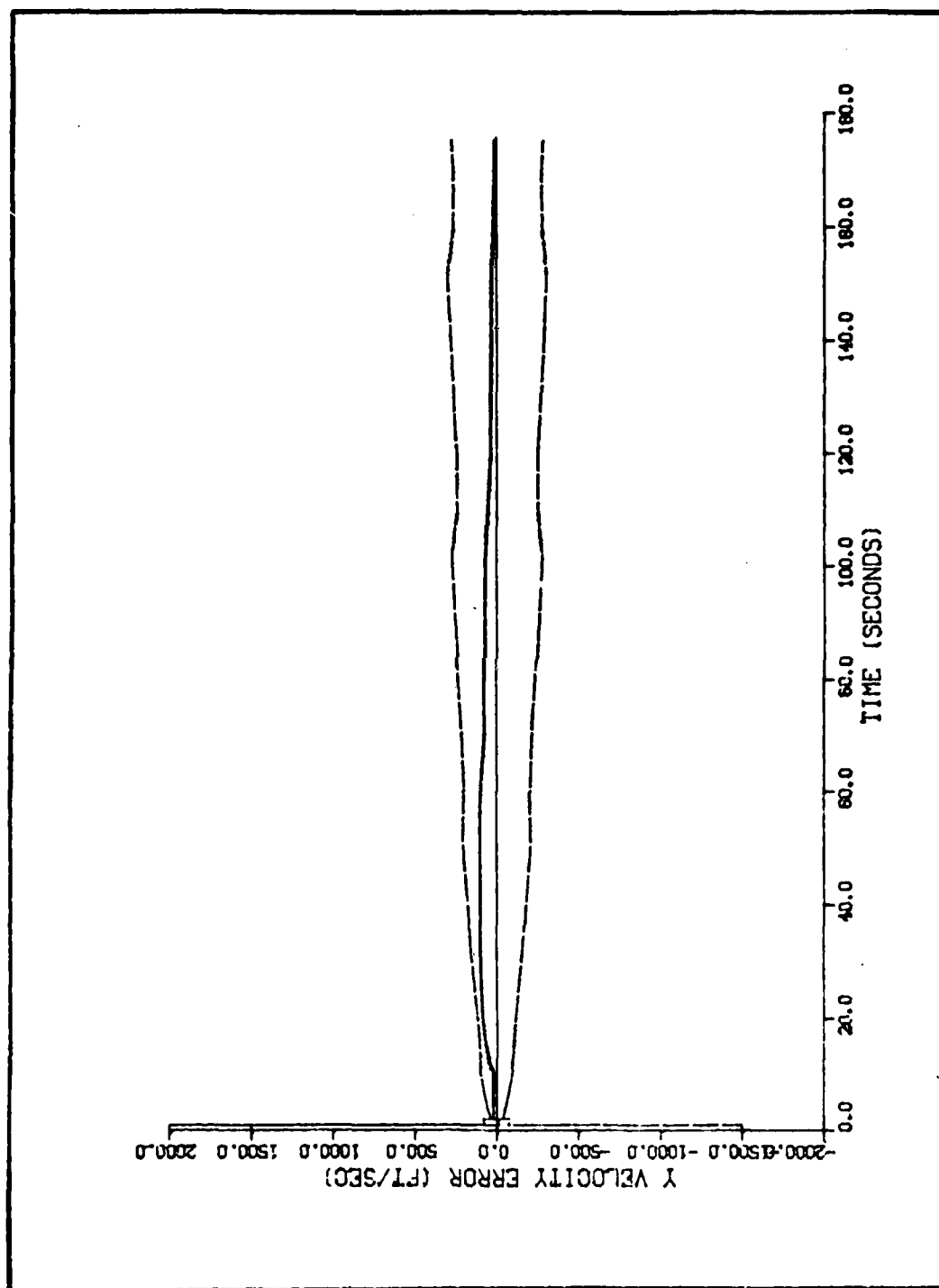


Figure O-4. Algorithm 5, Velocity Error in the Y Direction Versus Time,
Target Profile 5, $\sigma_{RATE} = 2 \text{ } \mu\text{rad/sec}$

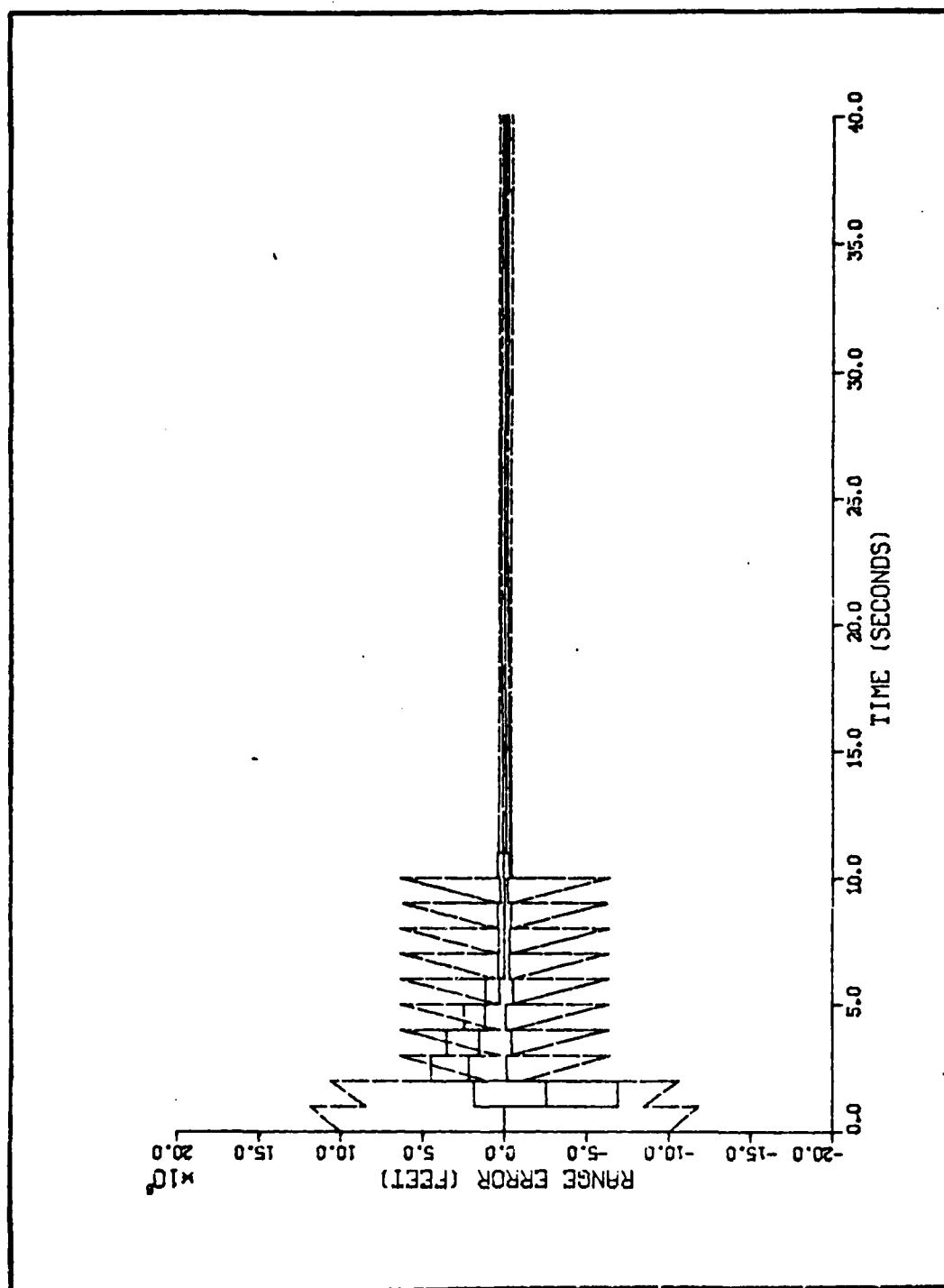


Figure 0-5. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\sigma_{\text{RATE}} = 200 \text{ } \mu\text{rad/sec}$

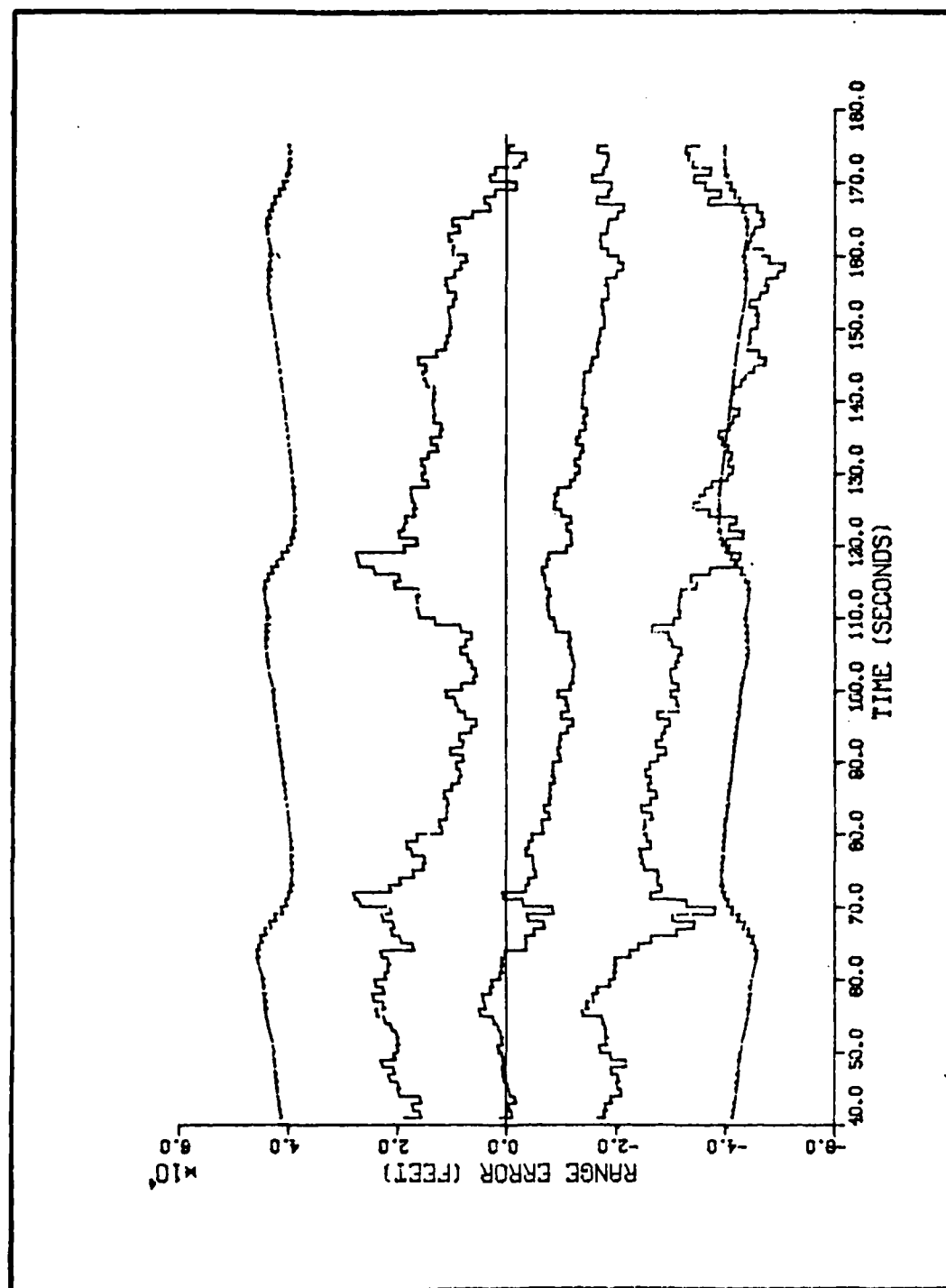


Figure O-6. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\sigma_{\text{RATE}} = 200 \text{ } \mu\text{rad/sec}$

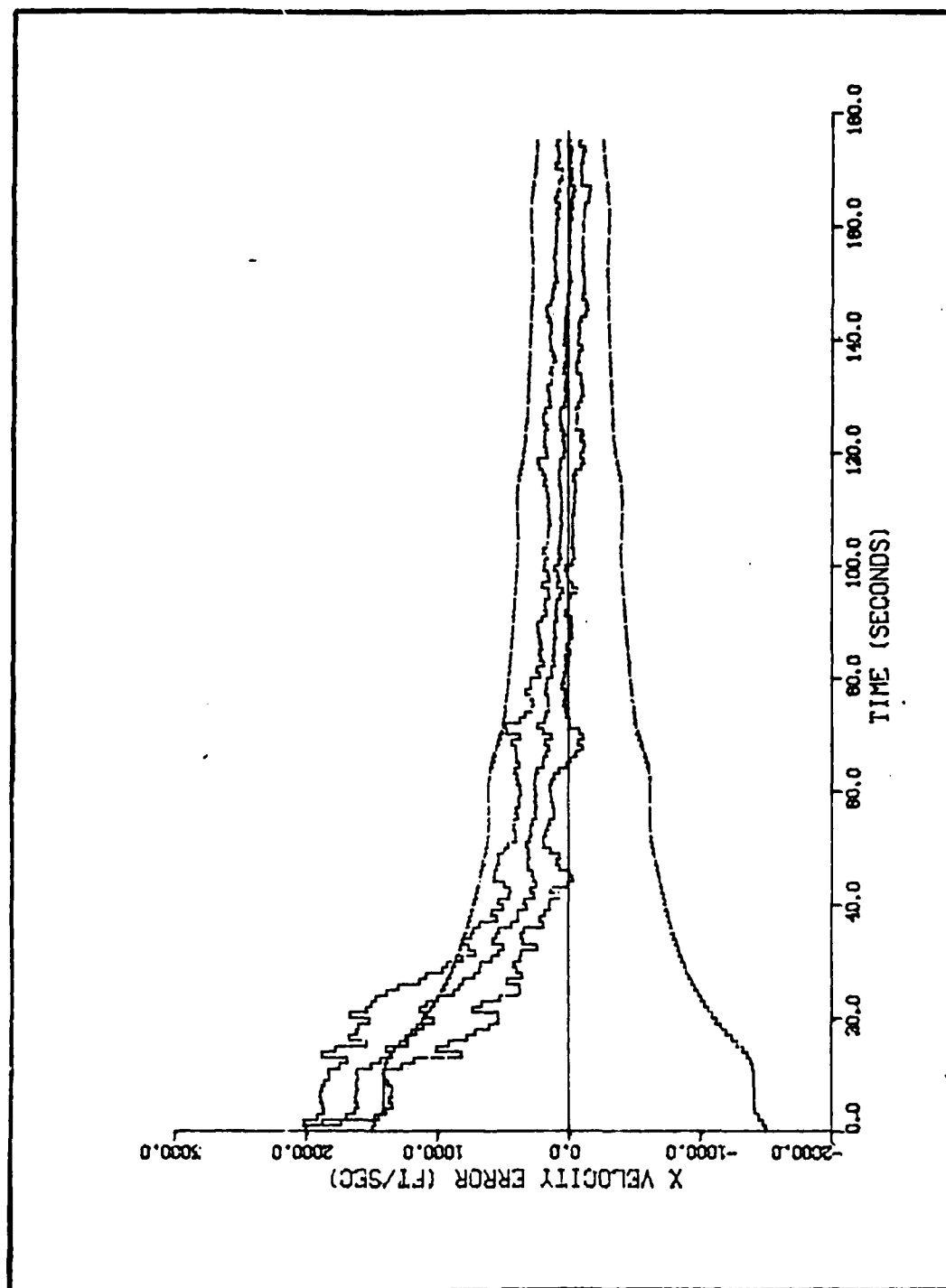


Figure O-7. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 5, $\sigma_{\text{RATE}} = 200 \text{ } \mu\text{rad/sec}$

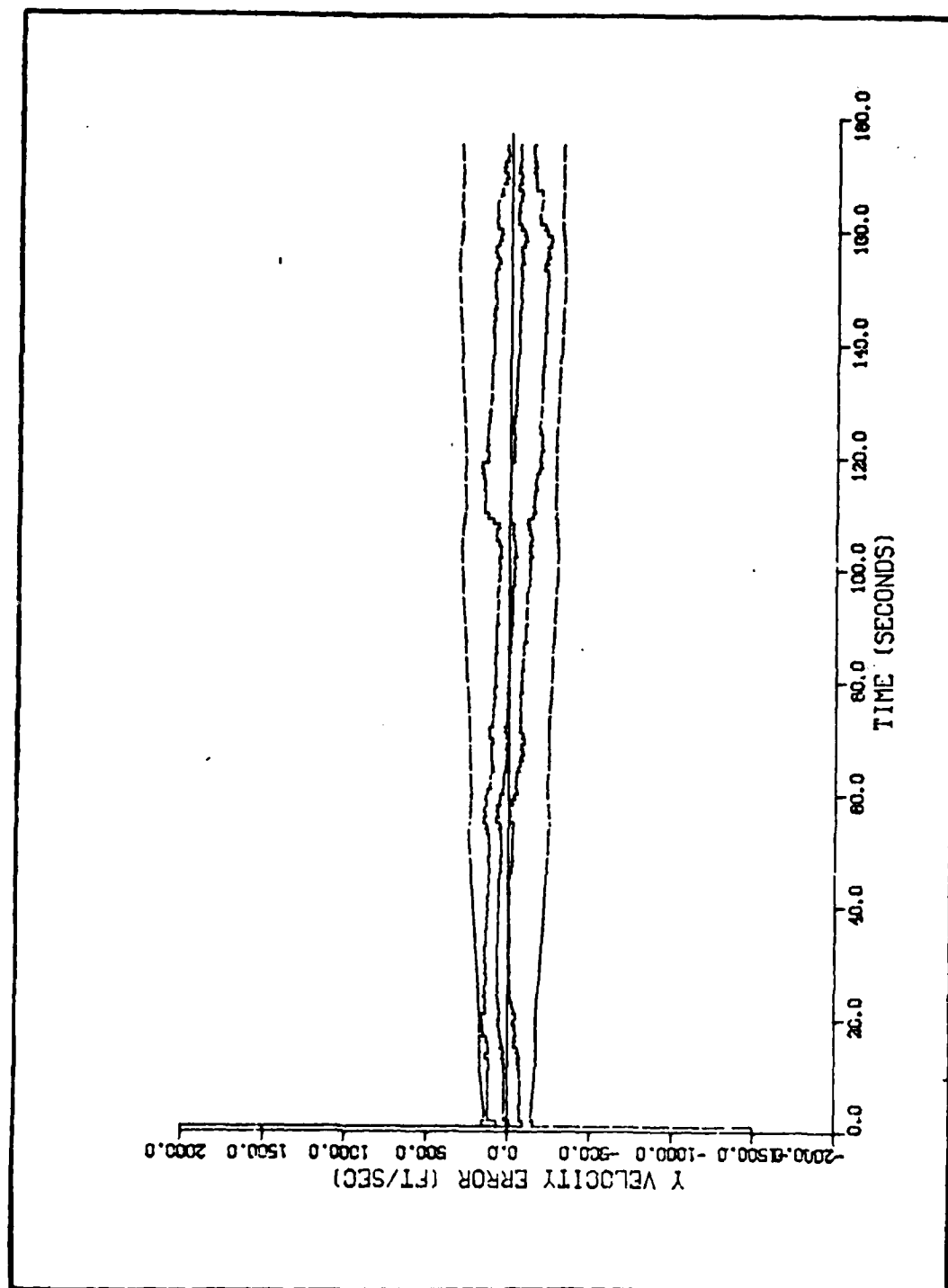


Figure 0-8. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 5, $\sigma_{\text{RATE}} = 200 \text{ } \mu\text{rad/sec}$

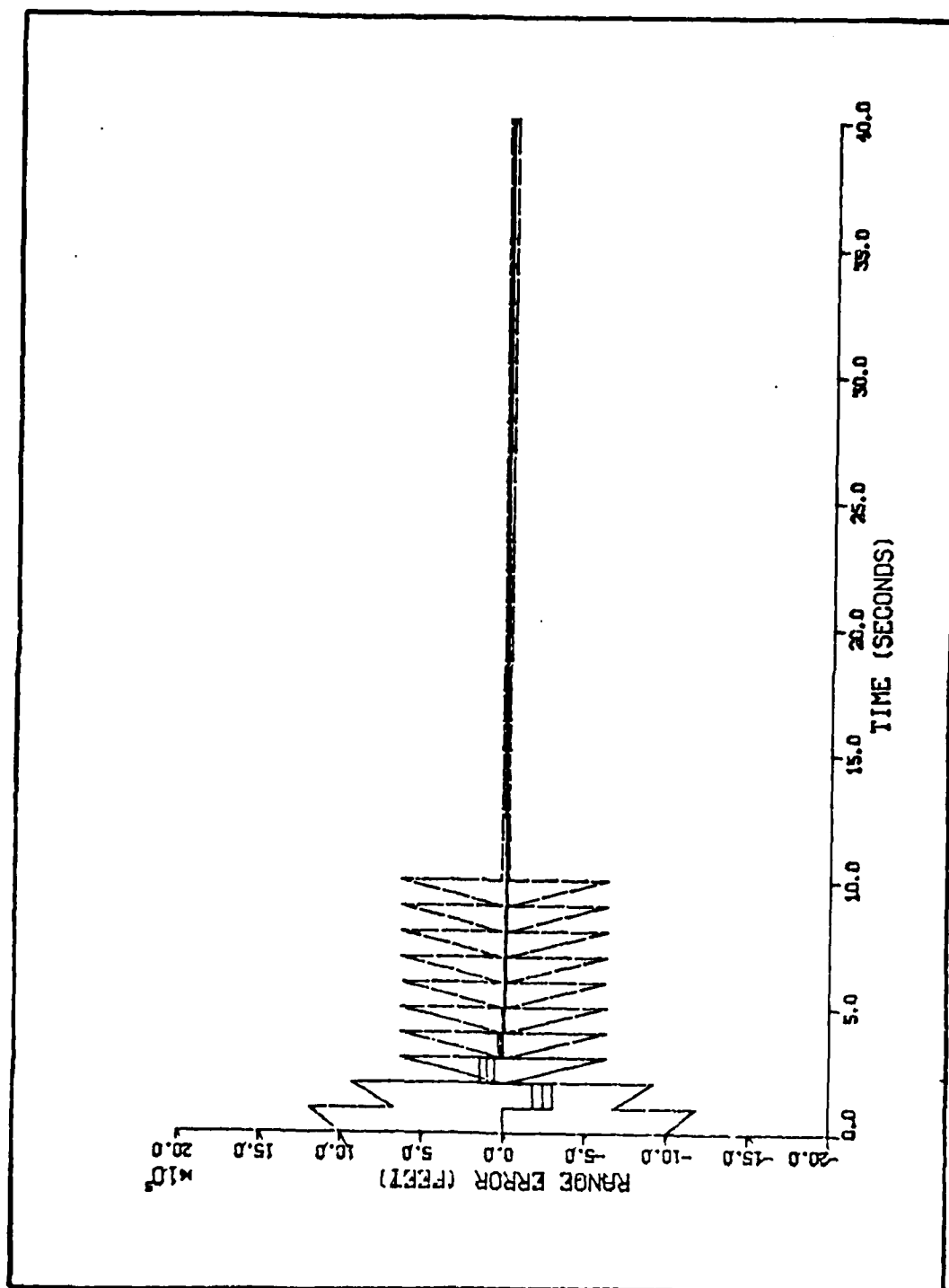


Figure 0-9. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\sigma_{\text{RATE}} = 10 \text{ } \mu\text{rad}$

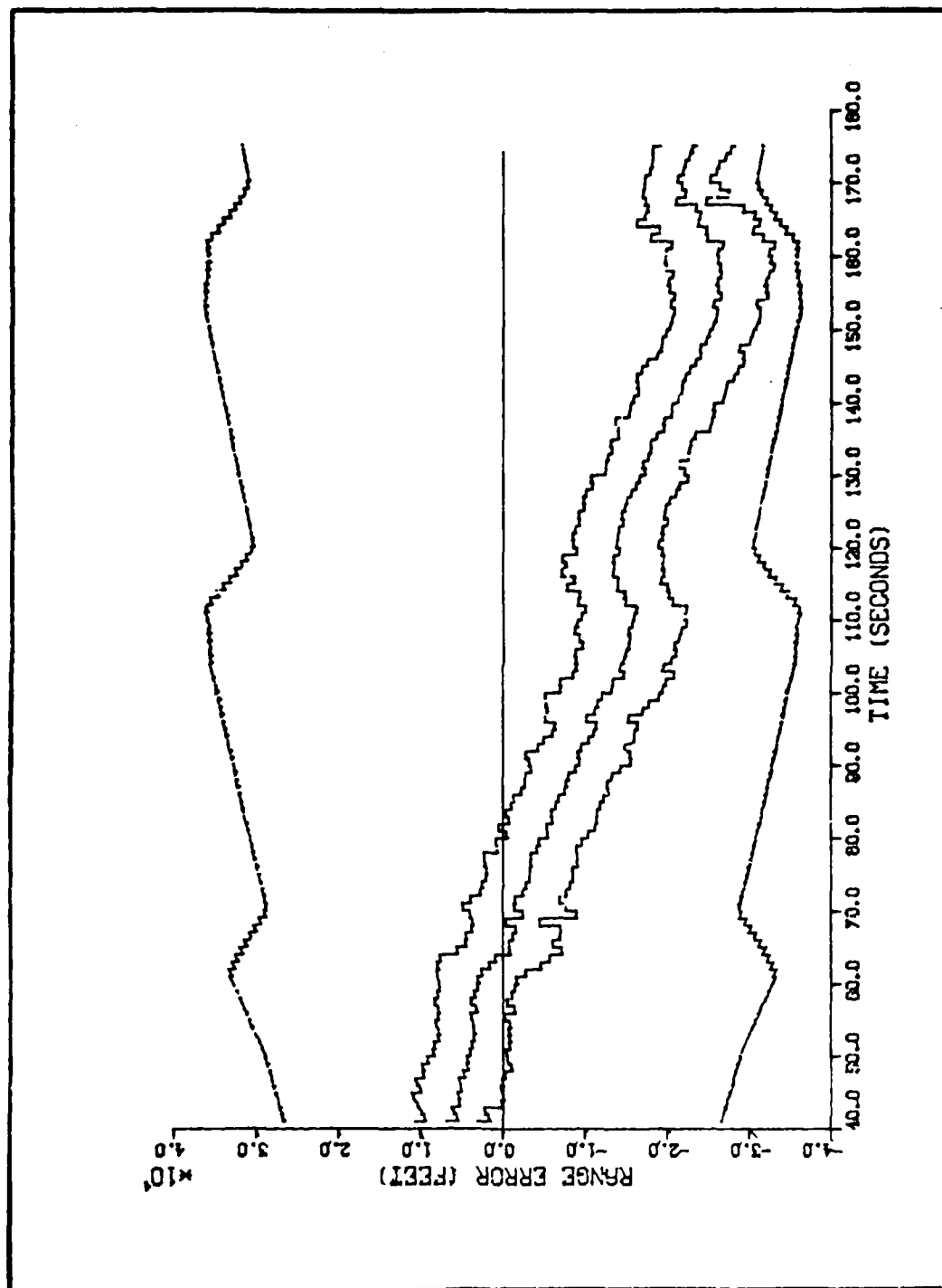


Figure 0-10. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\sigma_{\text{RATE}} = 10 \text{ } \mu\text{rad}$

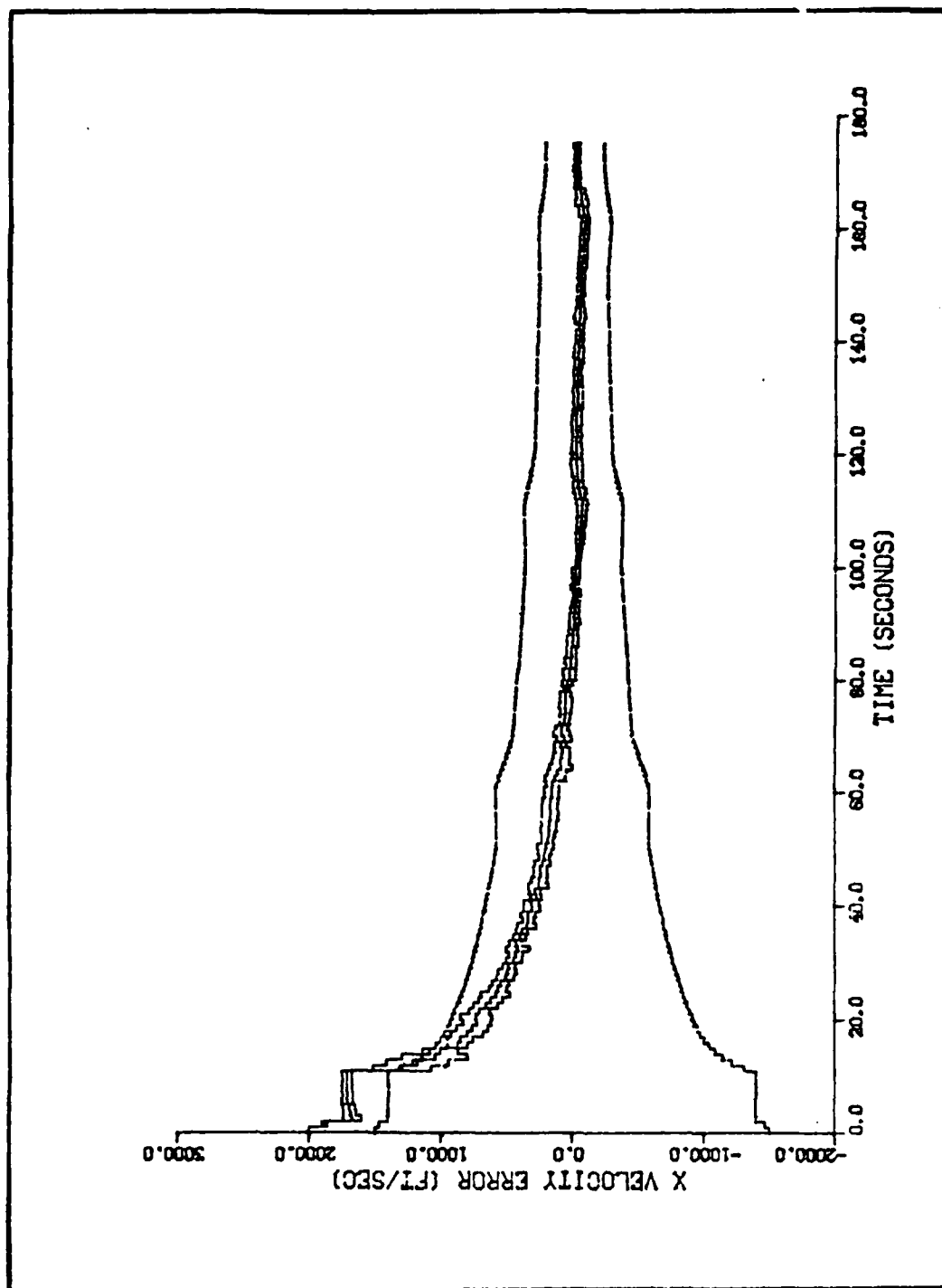


Figure O-11. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 5, $\sigma_{\text{ANGLE}} = 10 \text{ } \mu\text{rad}$

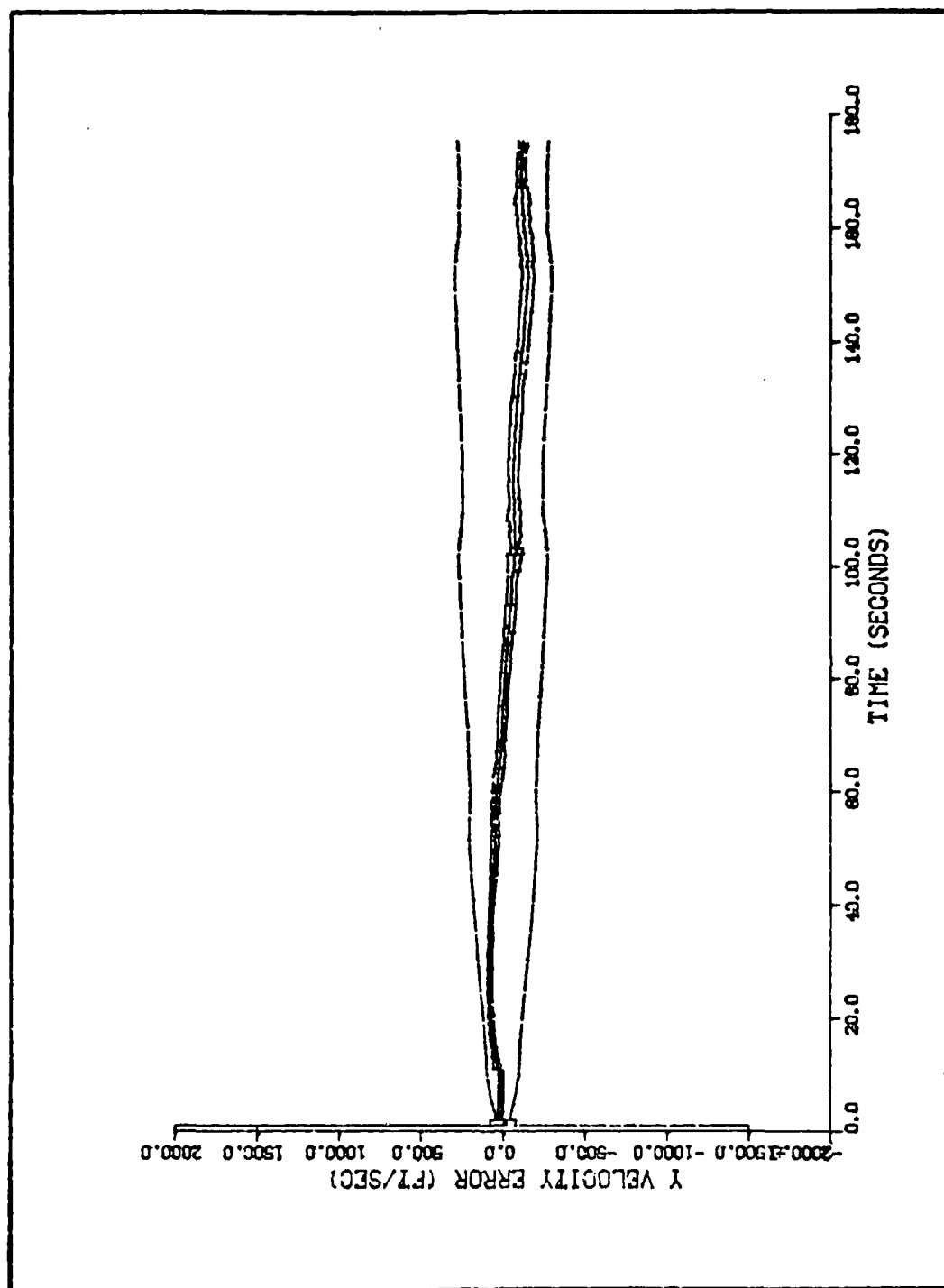


Figure O-12. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 5, $\sigma_{\text{ANGLE}} = 10 \text{ } \mu\text{rad}$

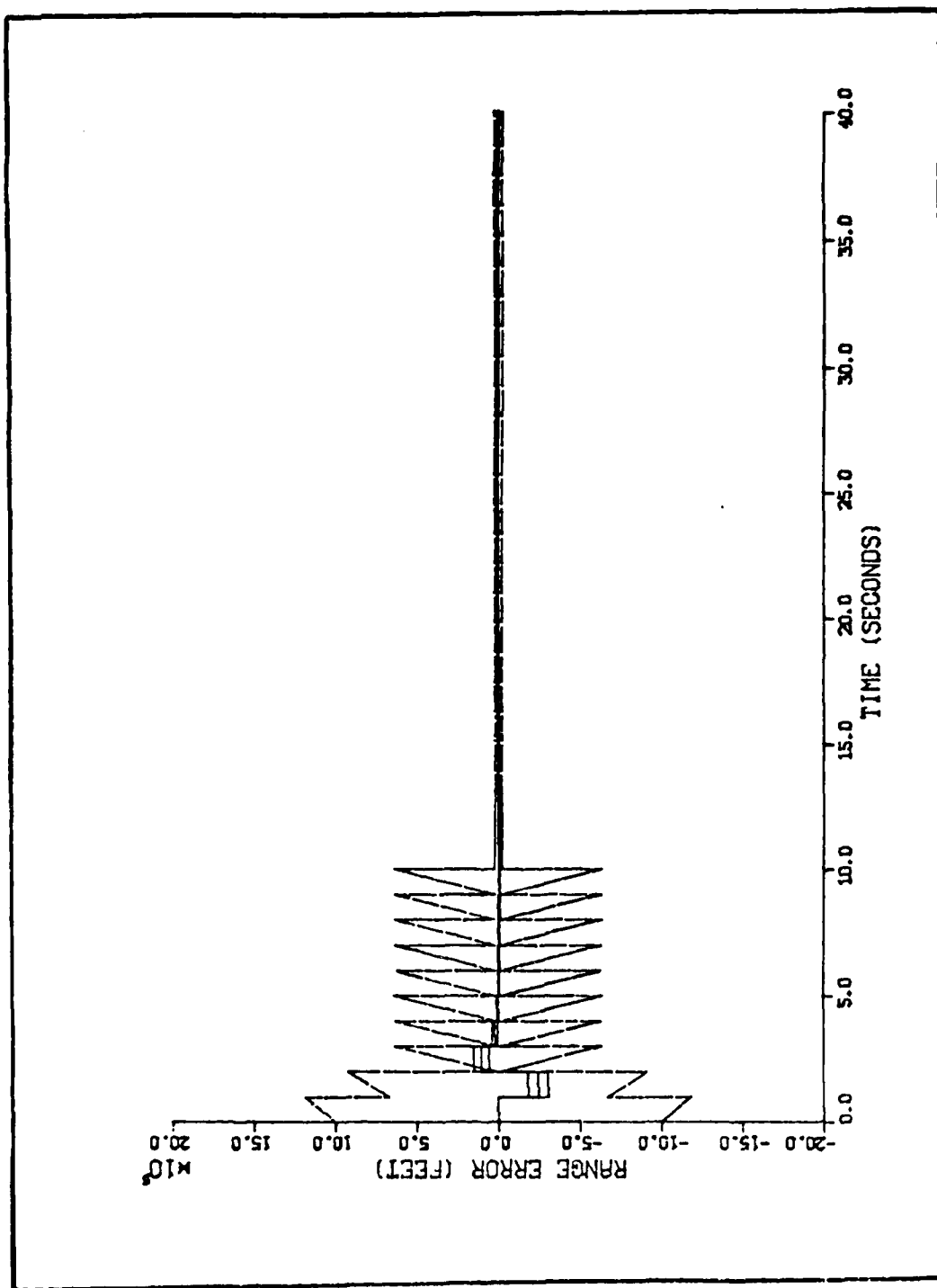


Figure O-13. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\sigma_{\text{ANGLE}} = 1 \text{ mrad}$

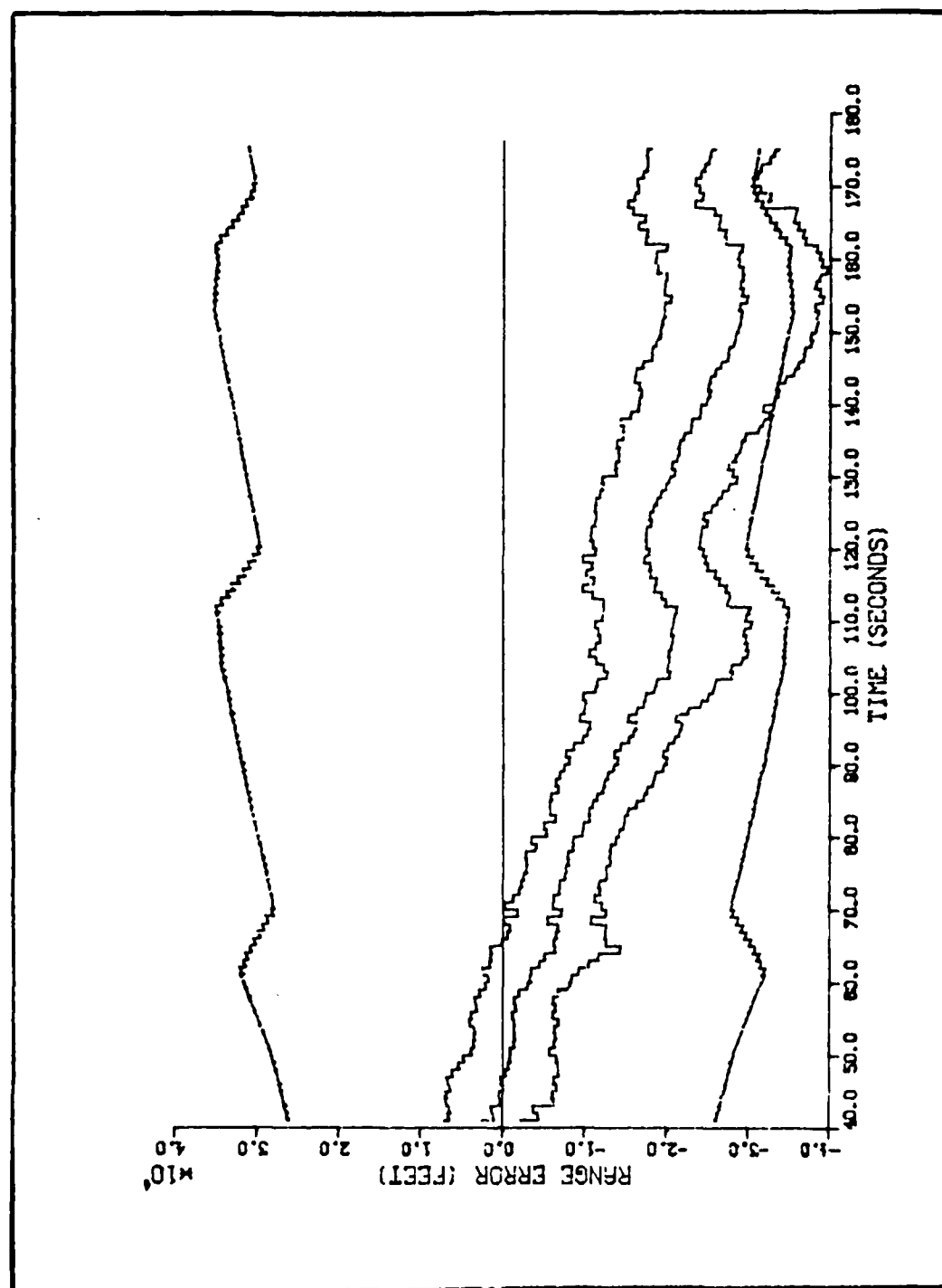


Figure C-14. Algorithm 5, Range Error Versus Time, Target Profile 5,
 $\theta_{\text{ANGLE}} = 1 \text{ mrad}$

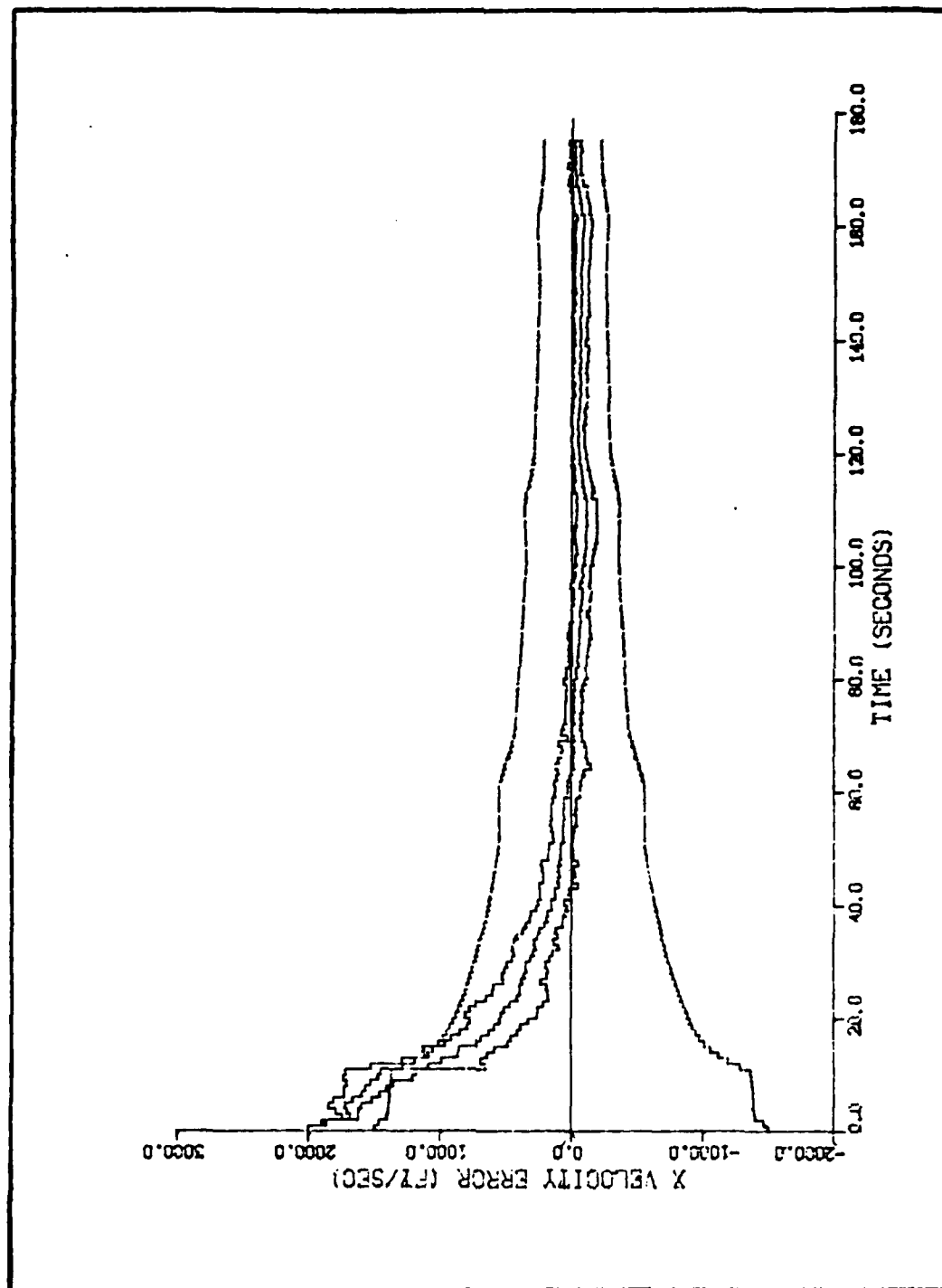


Figure O-15. Algorithm 5, Velocity Error in the X Direction Versus Time,
Target Profile 5, $\sigma_{\text{ANGLE}} = 1 \text{ mrad}$

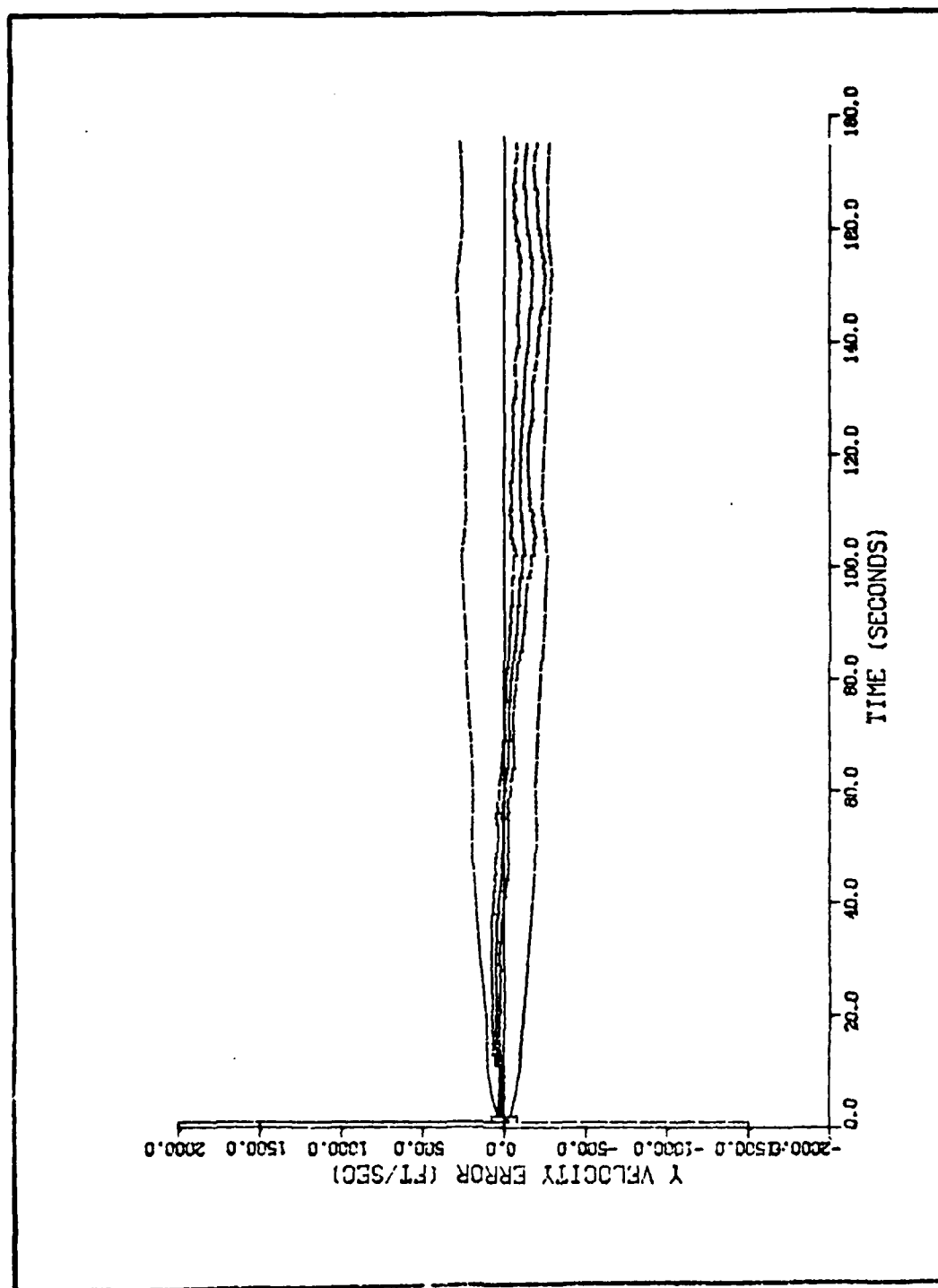


Figure O-16. Algorithm 5, Velocity Error in the Y Direction Versus Time,
Target Profile 5, $\sigma_{\text{ANGLE}} = 1 \text{ mrad}$

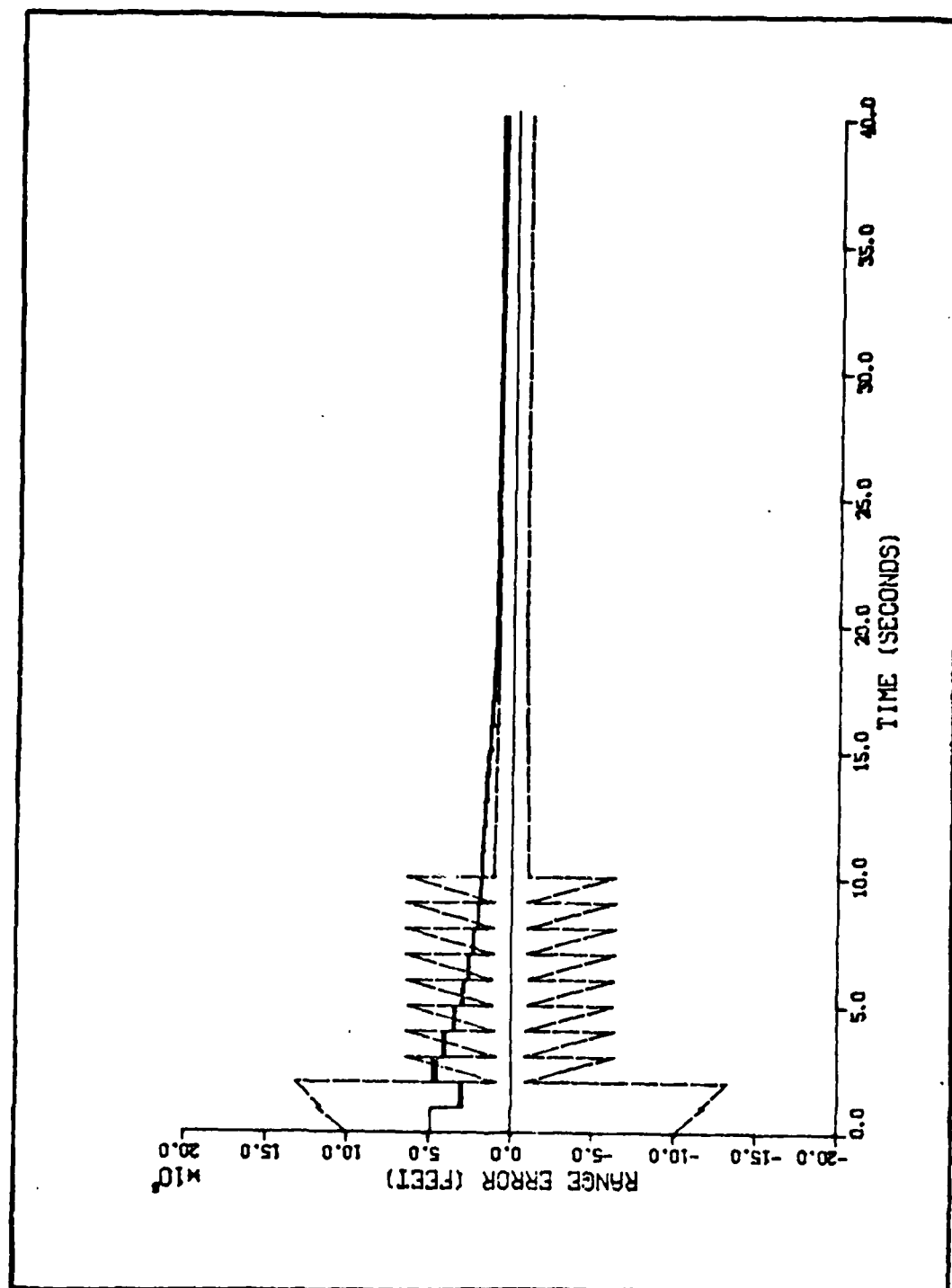


Figure 0-17. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\sigma_{\text{RATE}} = 2 \text{ urad/sec}$

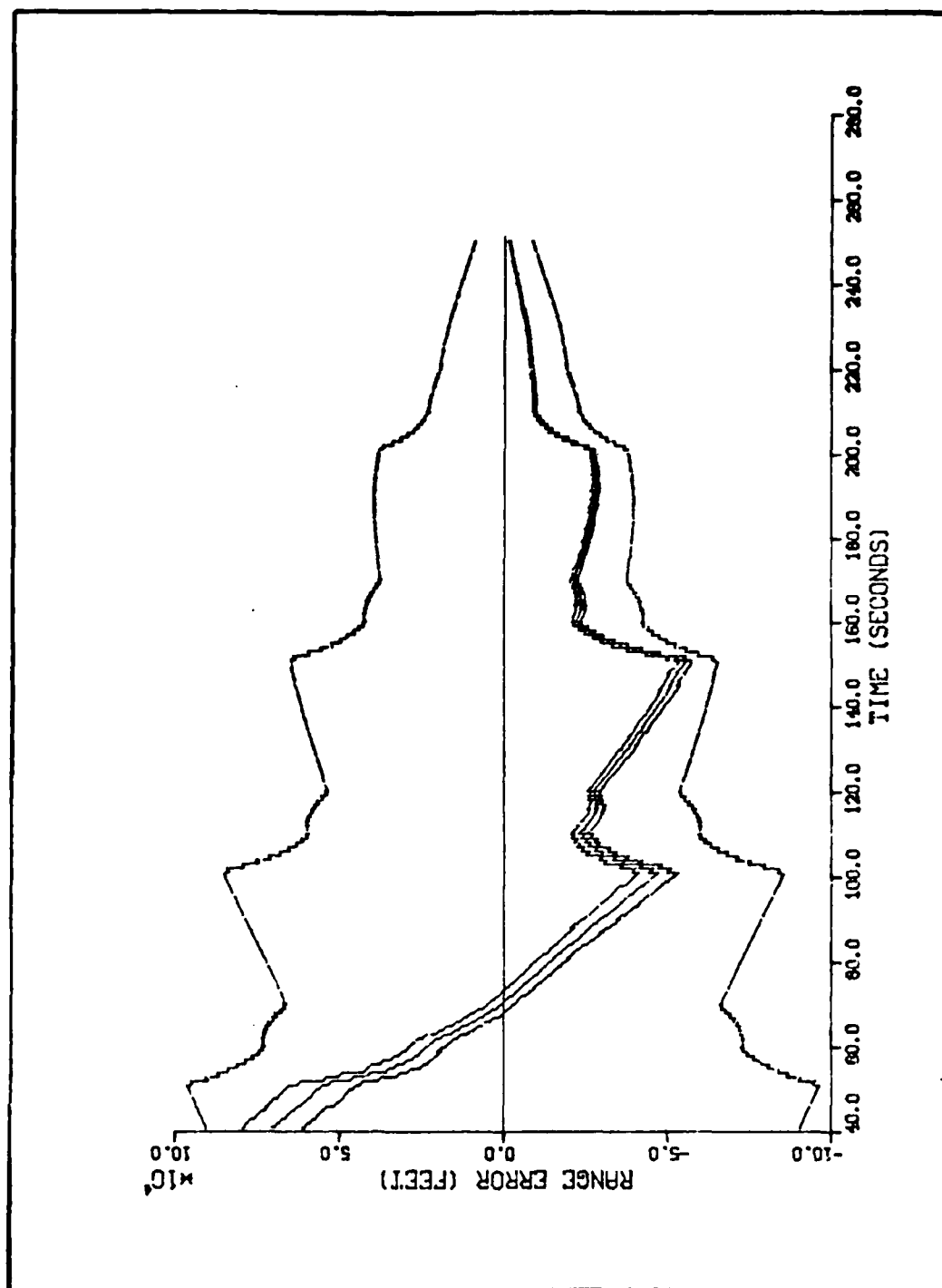


Figure O-18. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\sigma_{\text{RATE}} = 2 \text{ } \mu\text{rad/sec}$

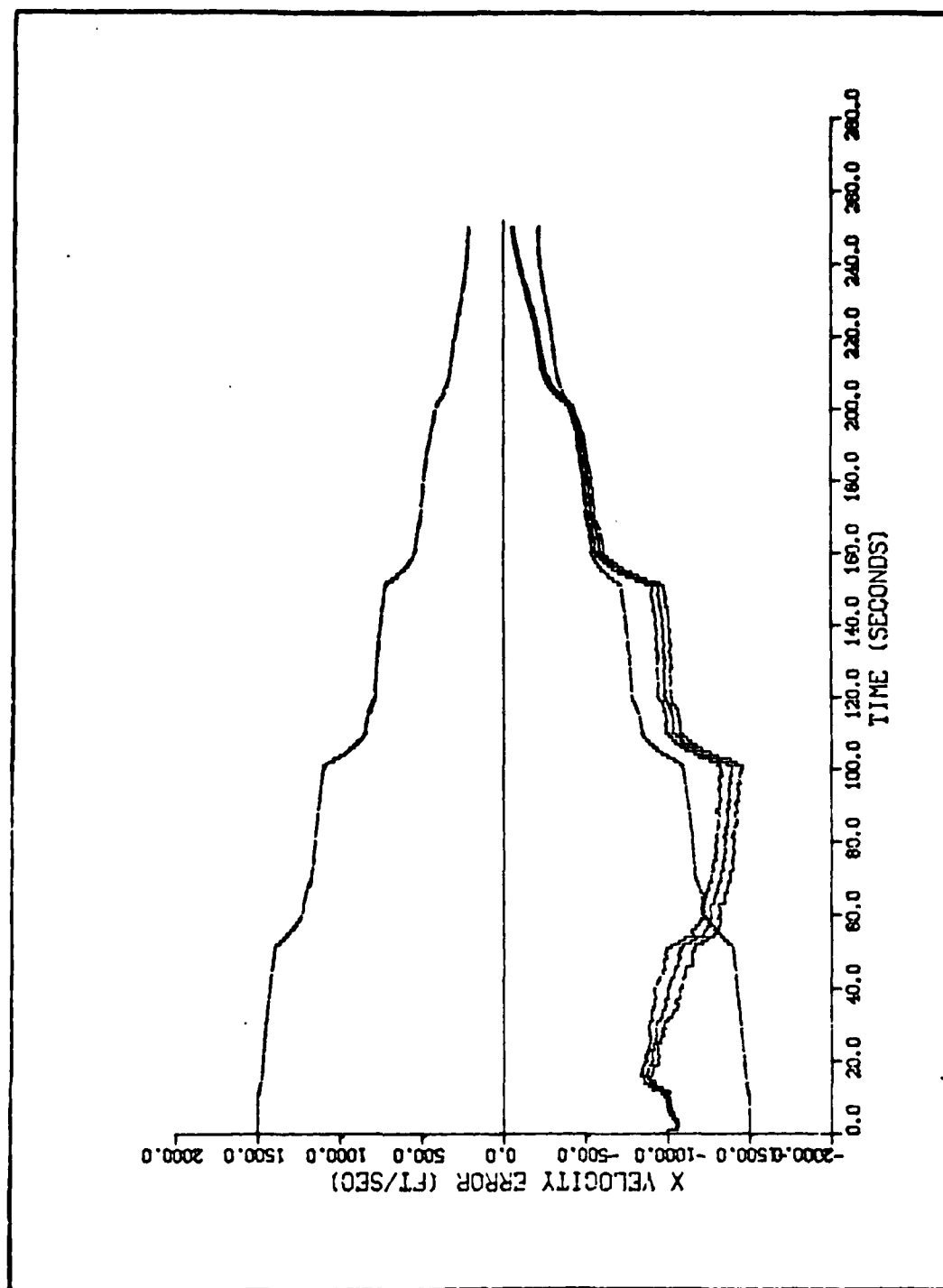


Figure 0-19. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 7, $\sigma_{\text{RATE}} = 2 \text{ rad/sec}$

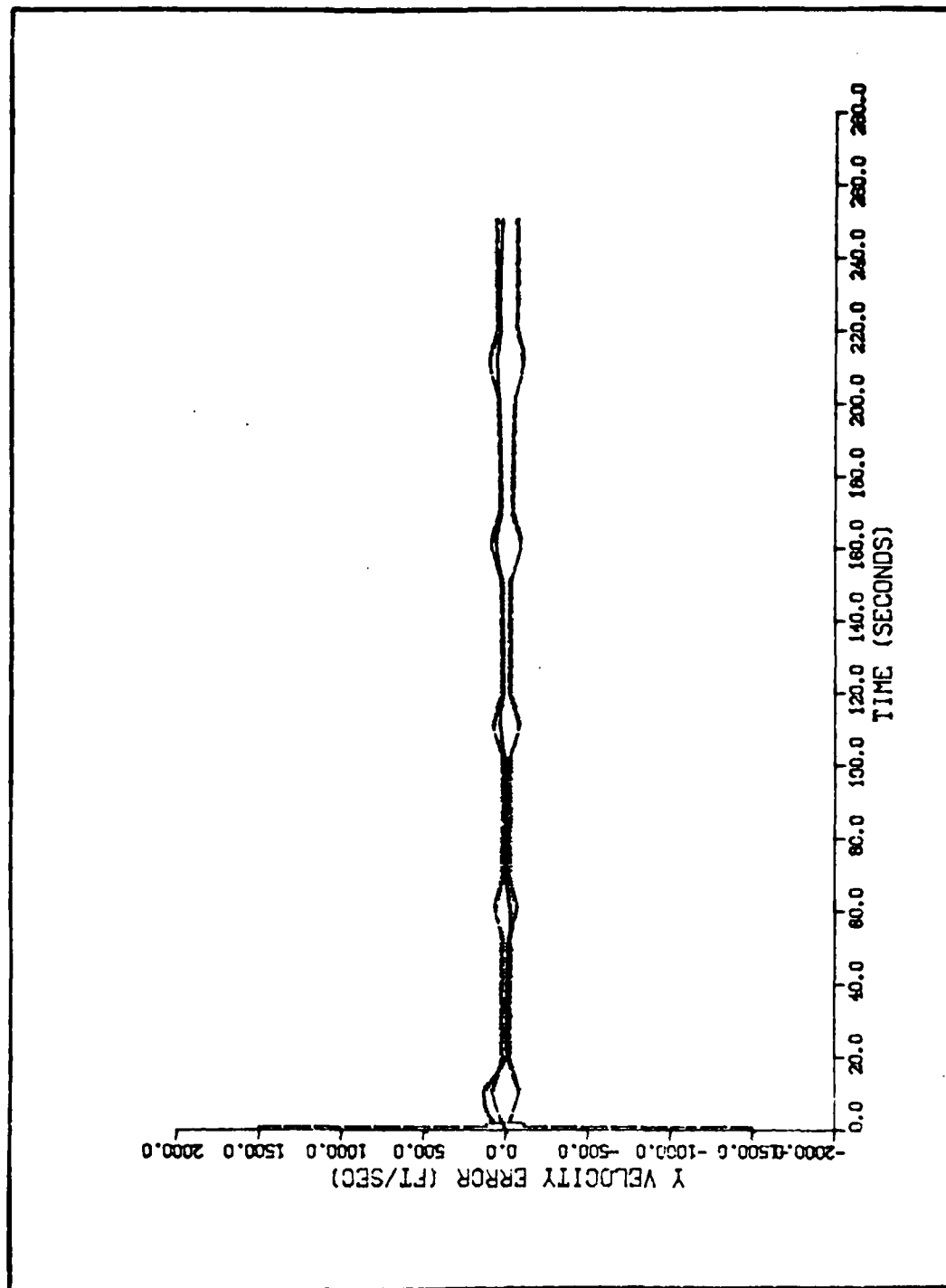


Figure O-20. Algorithm 5, Velocity Error in the Y Direction Versus Time, Target Profile 7, $\sigma_{\text{RATE}} = 2 \text{ rad/sec}$

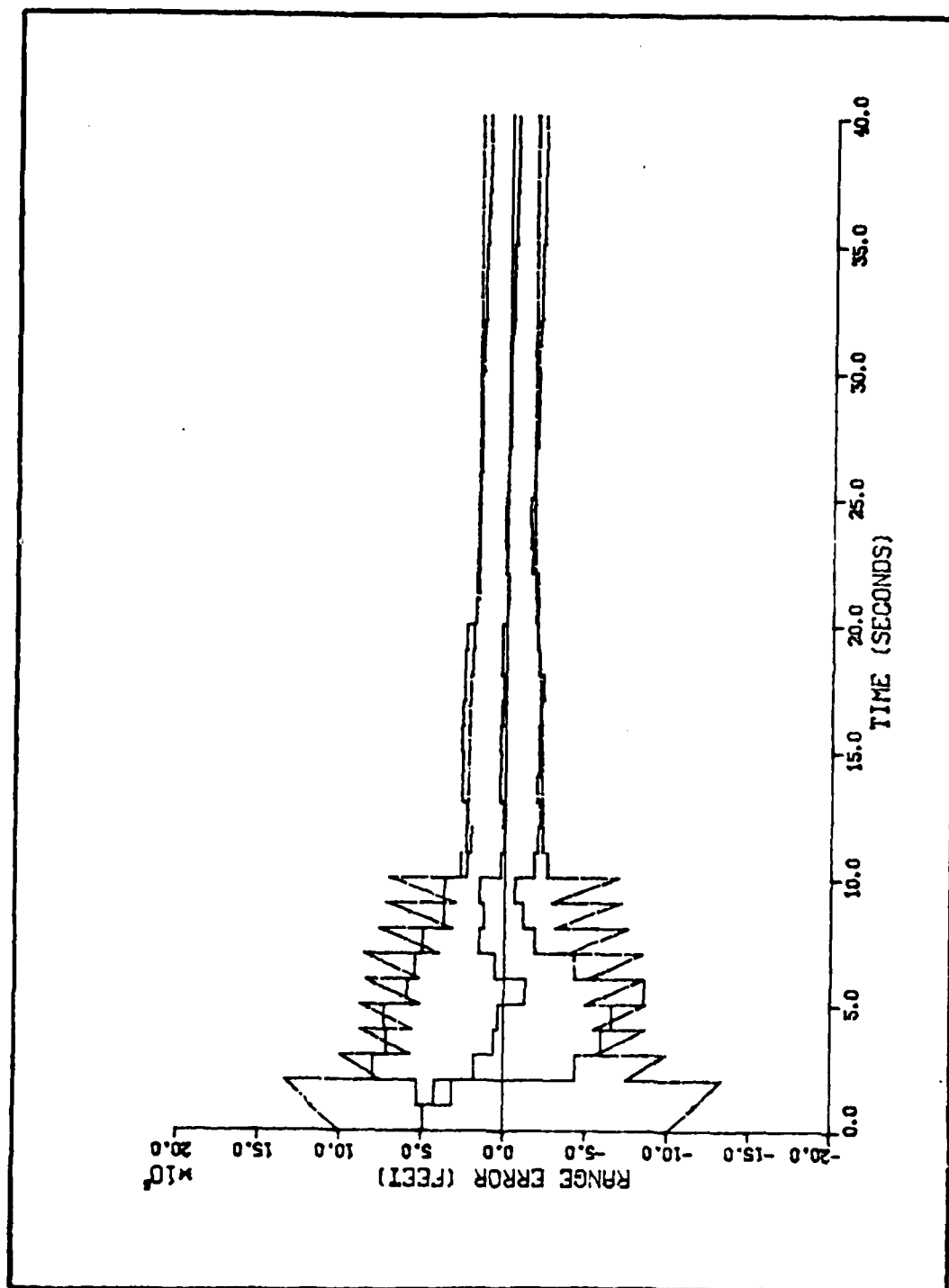


Figure O-21. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\theta_{\text{RATE}} = 200 \text{ } \mu\text{rad/sec}$

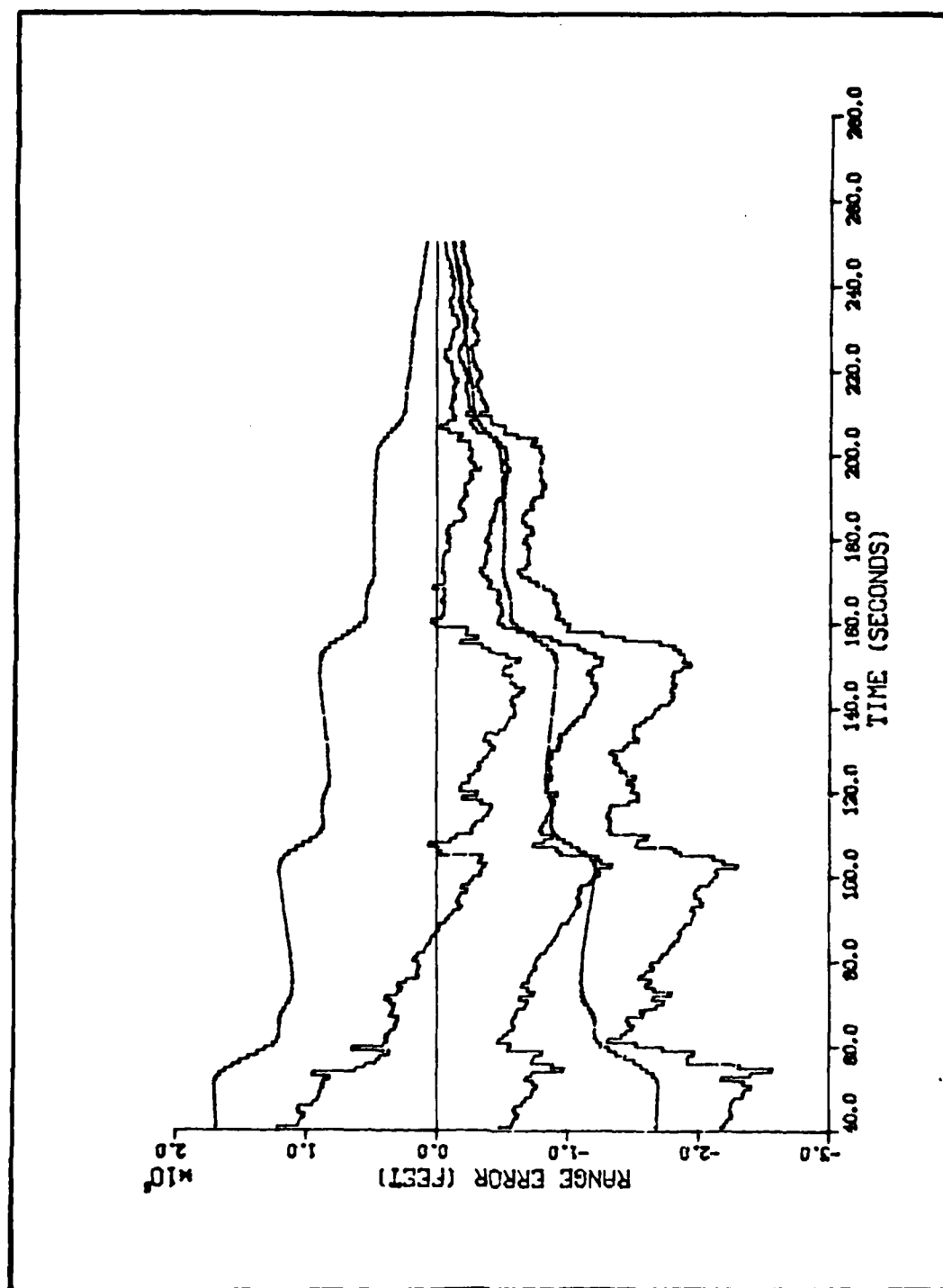


Figure O-22. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\omega_{\text{RATE}} = 200 \text{ } \mu\text{rad/sec}$

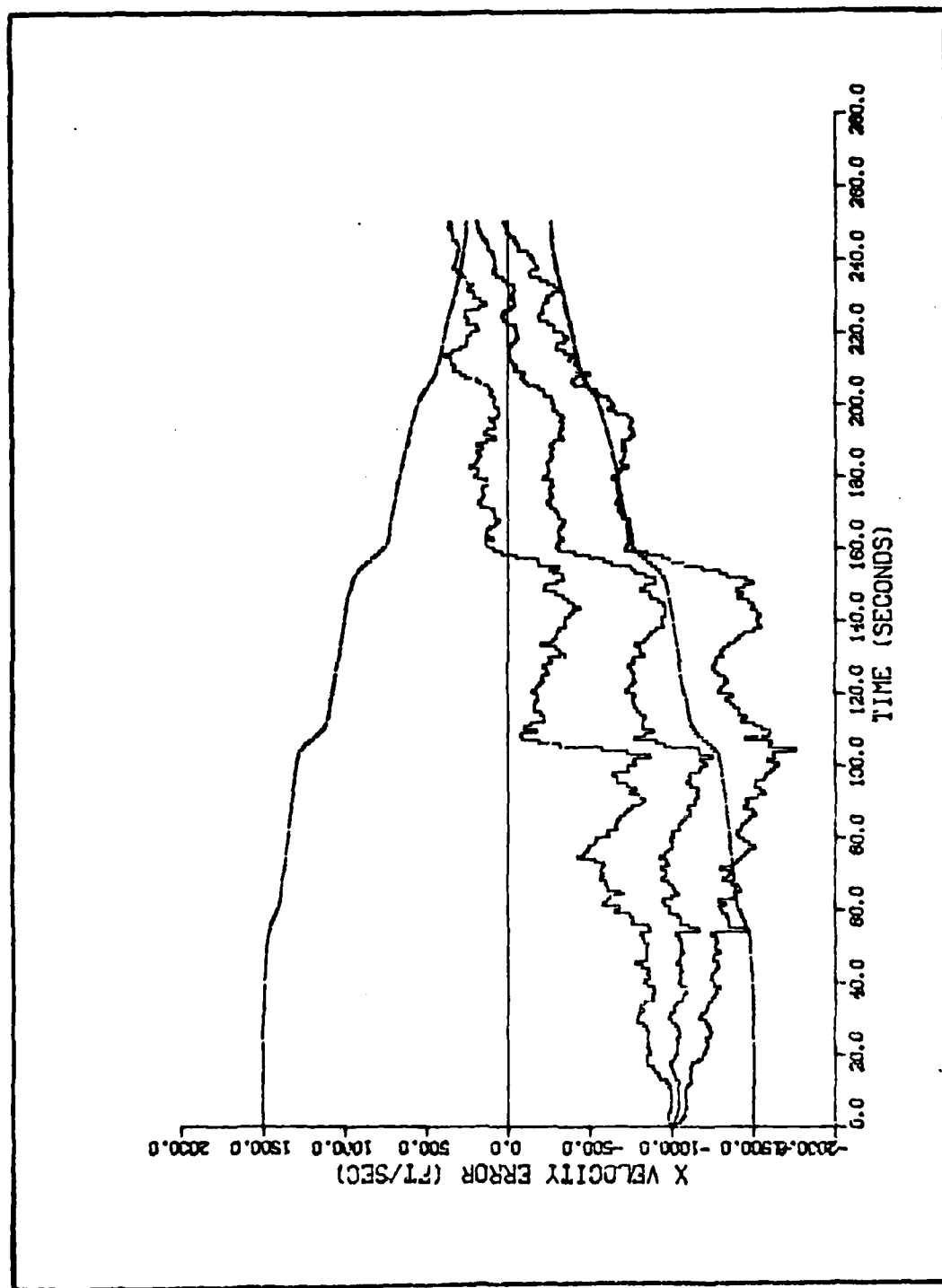


Figure O-23. Algorithm 5, Velocity Error in the X Direction Versus Time,
Target Profile 7, °RATE = 200 μrad/sec

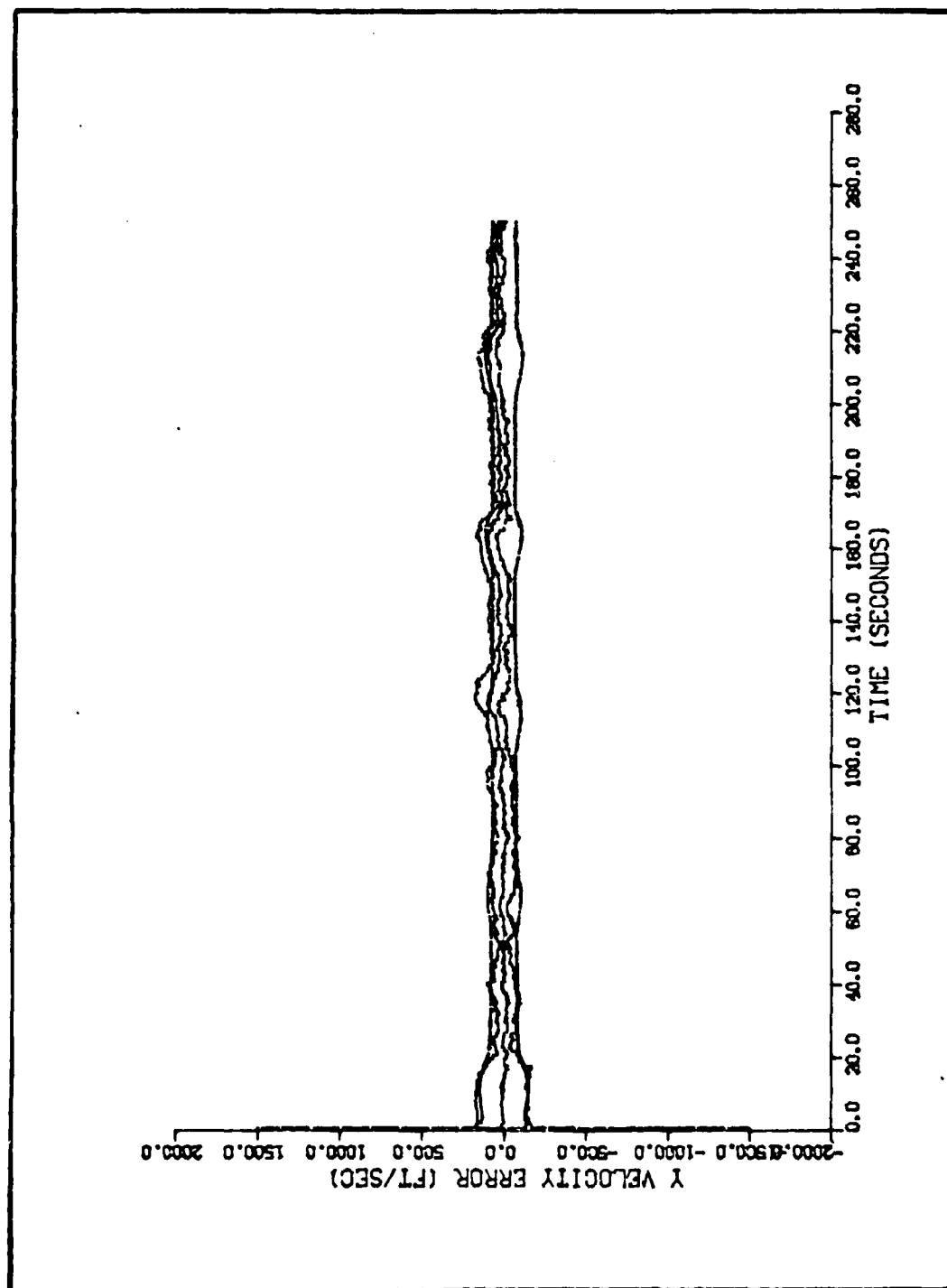


Figure O-24. Algorithm 5, Velocity Error in the Y direction Versus Time,
Target Profile 7, $\sigma_{\text{RATE}} = 200 \text{ } \mu\text{rad/sec}$

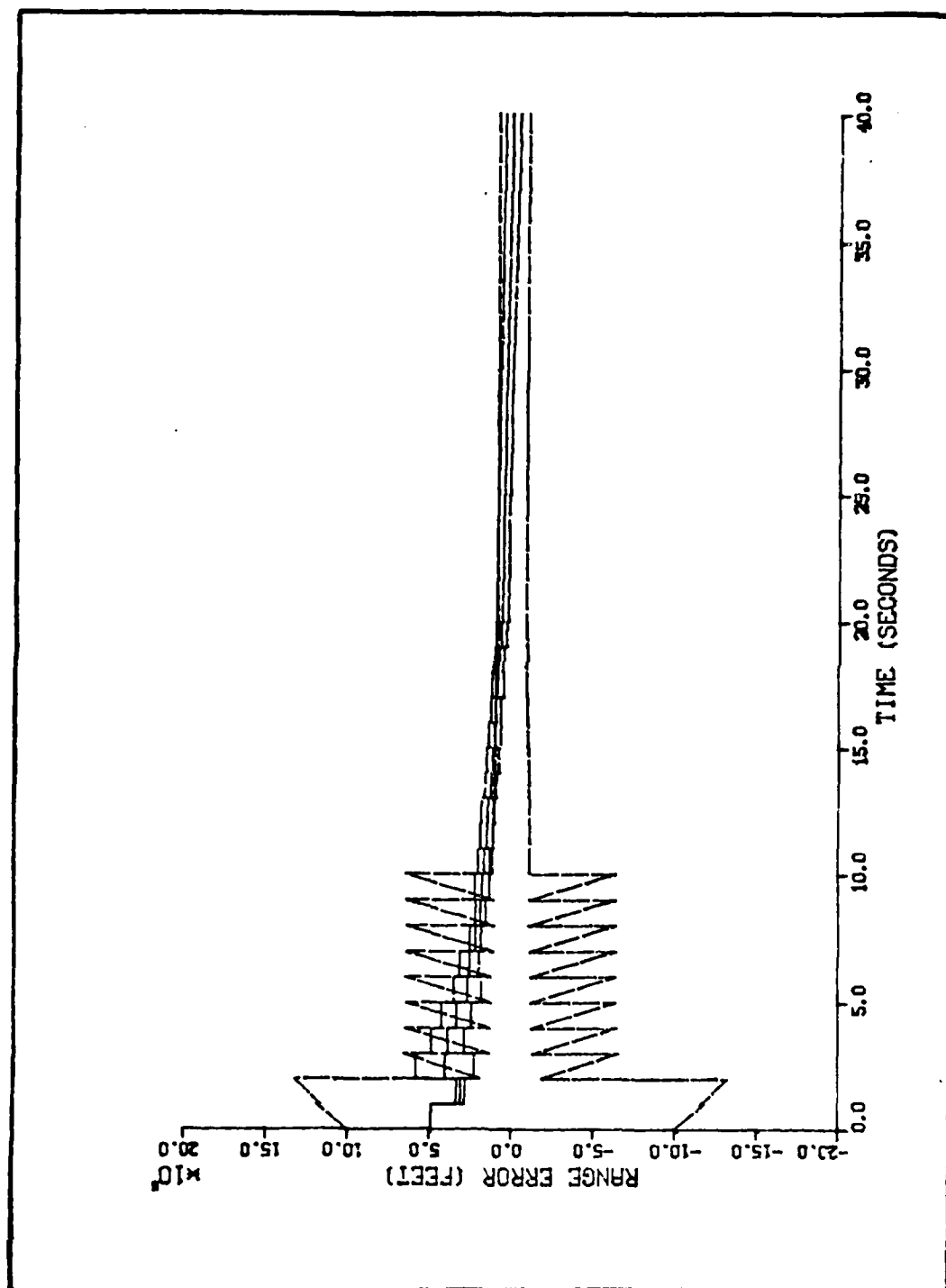


Figure 0-25. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\sigma_{\text{ANGLE}} = 10 \text{ } \mu\text{rad}$

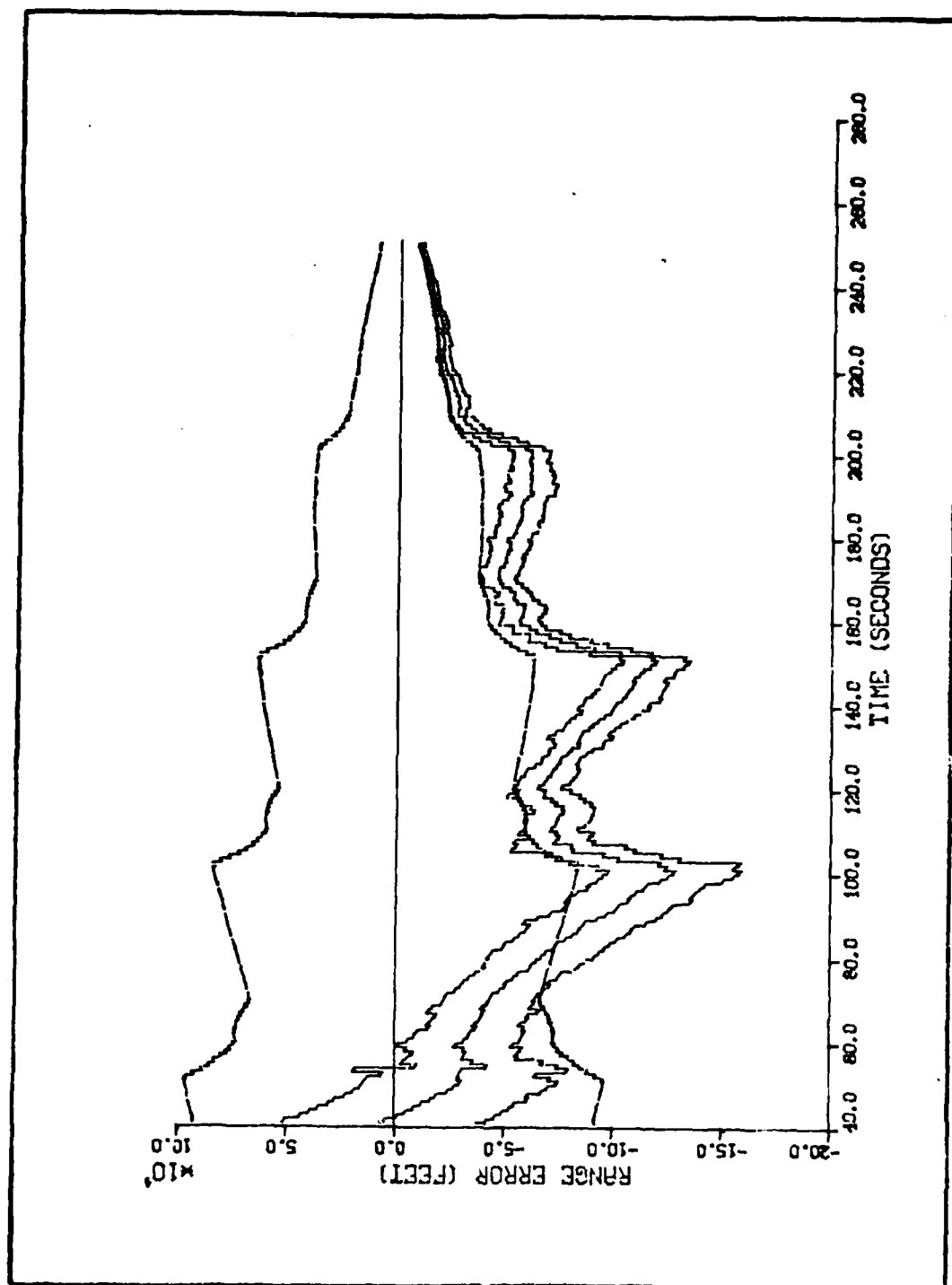


Figure O-26. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\sigma_{\text{ANGLE}} = 10 \text{ } \mu\text{rad}$

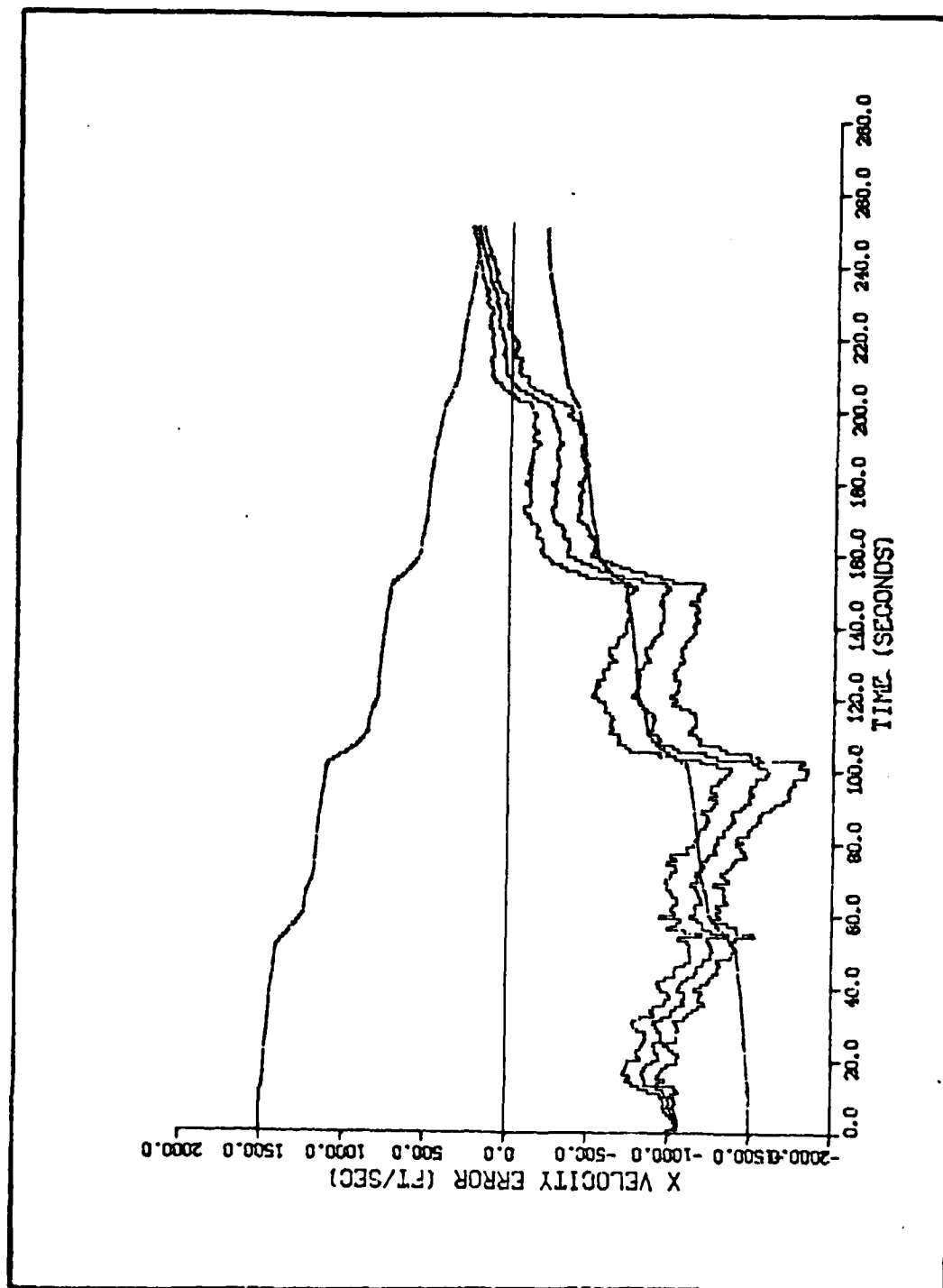


Figure O-27. Algorithm 5, Velocity Error in the X Direction Versus Time, Target Profile 7, $\epsilon = 10 \mu\text{rad}$

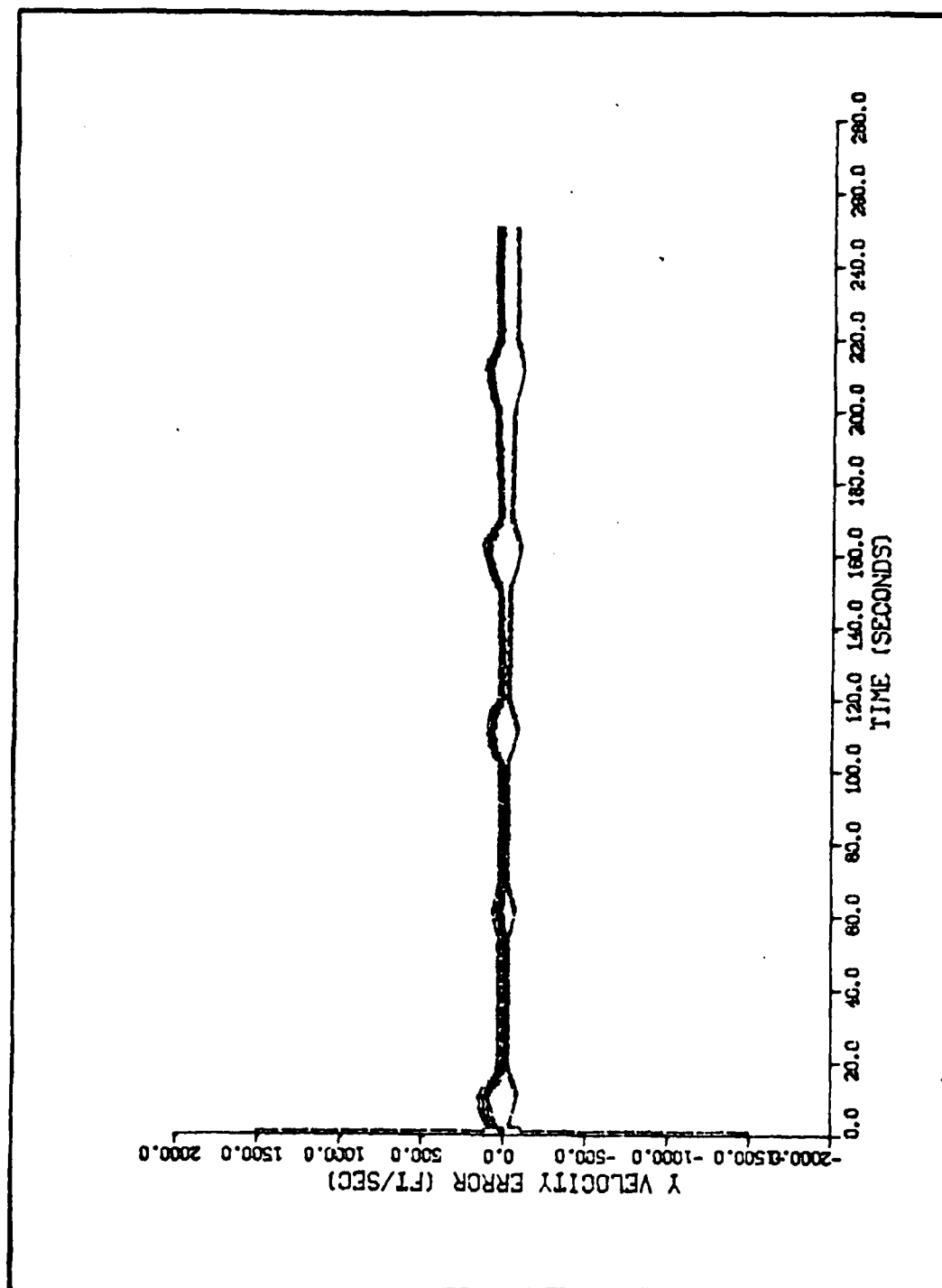


Figure O-28. Algorithm 5, Velocity Error in the Y Direction Versus Time,
Target Profile 7, σ ANGLE = 10 μ rad

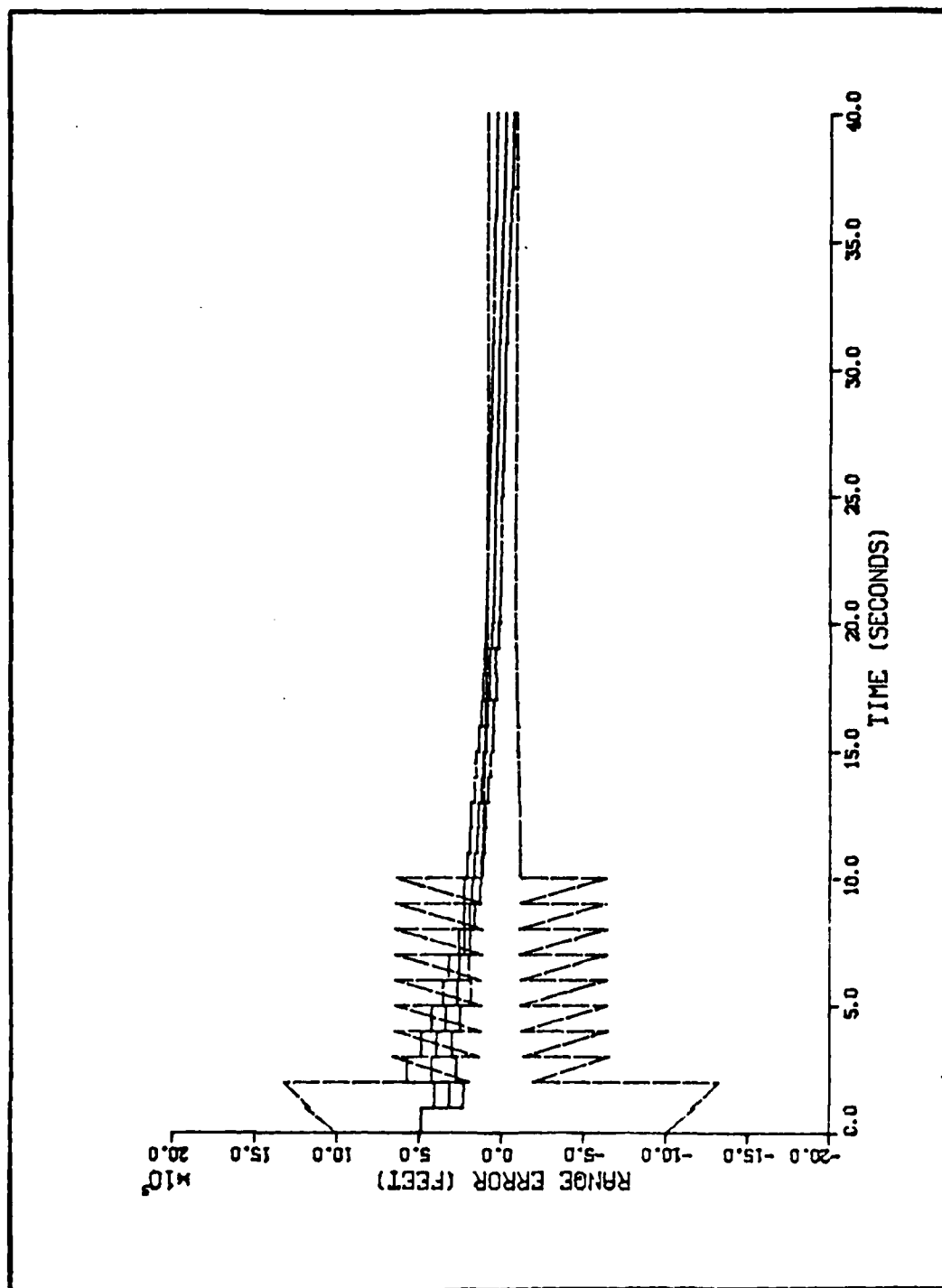


Figure O-29. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\sigma_{\text{ANGLE}} = 1 \text{ mrad}$

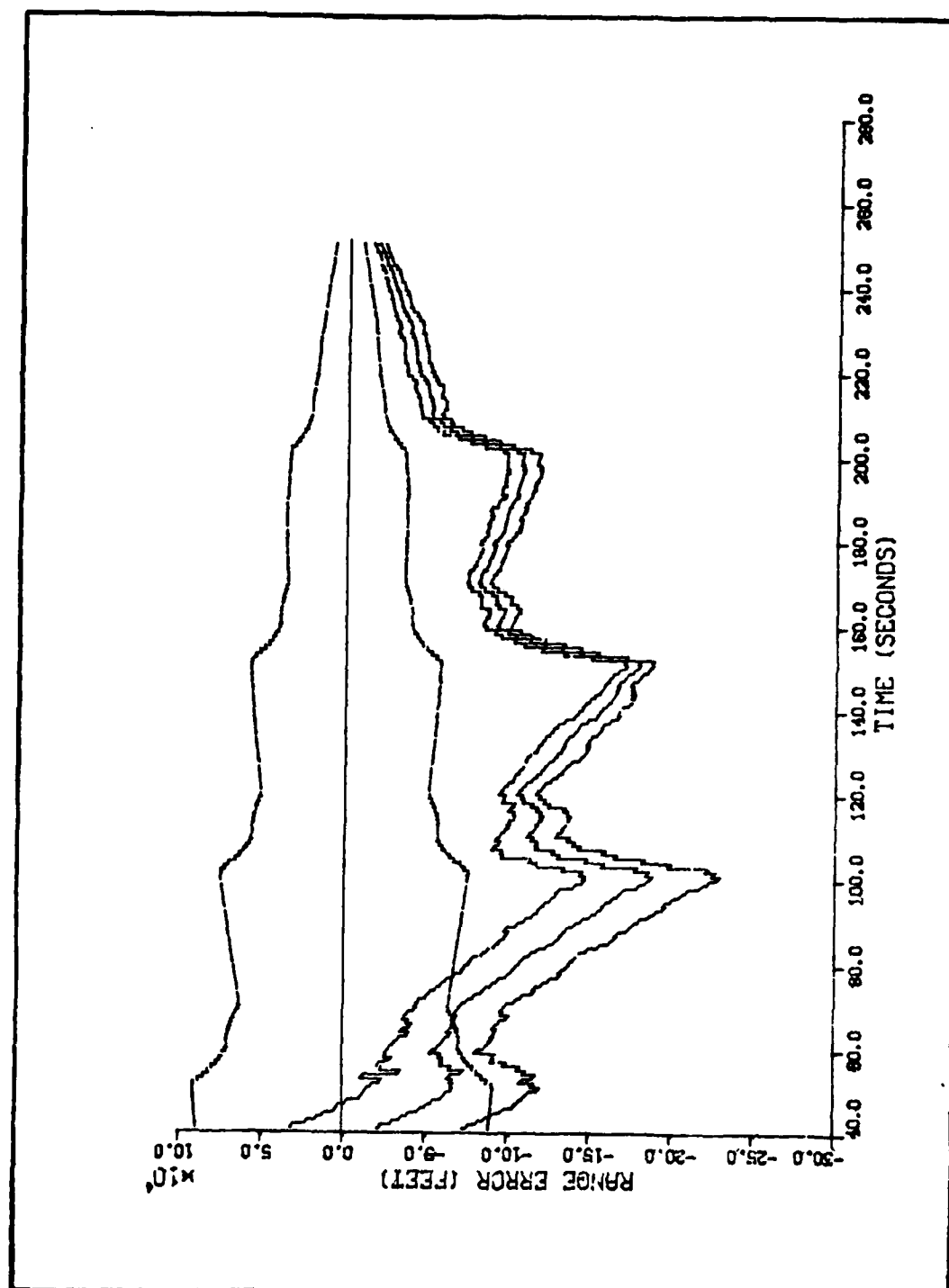


Figure 0-30. Algorithm 5, Range Error Versus Time, Target Profile 7,
 $\sigma_{\text{ANGLE}} = 1 \text{ mrad}$

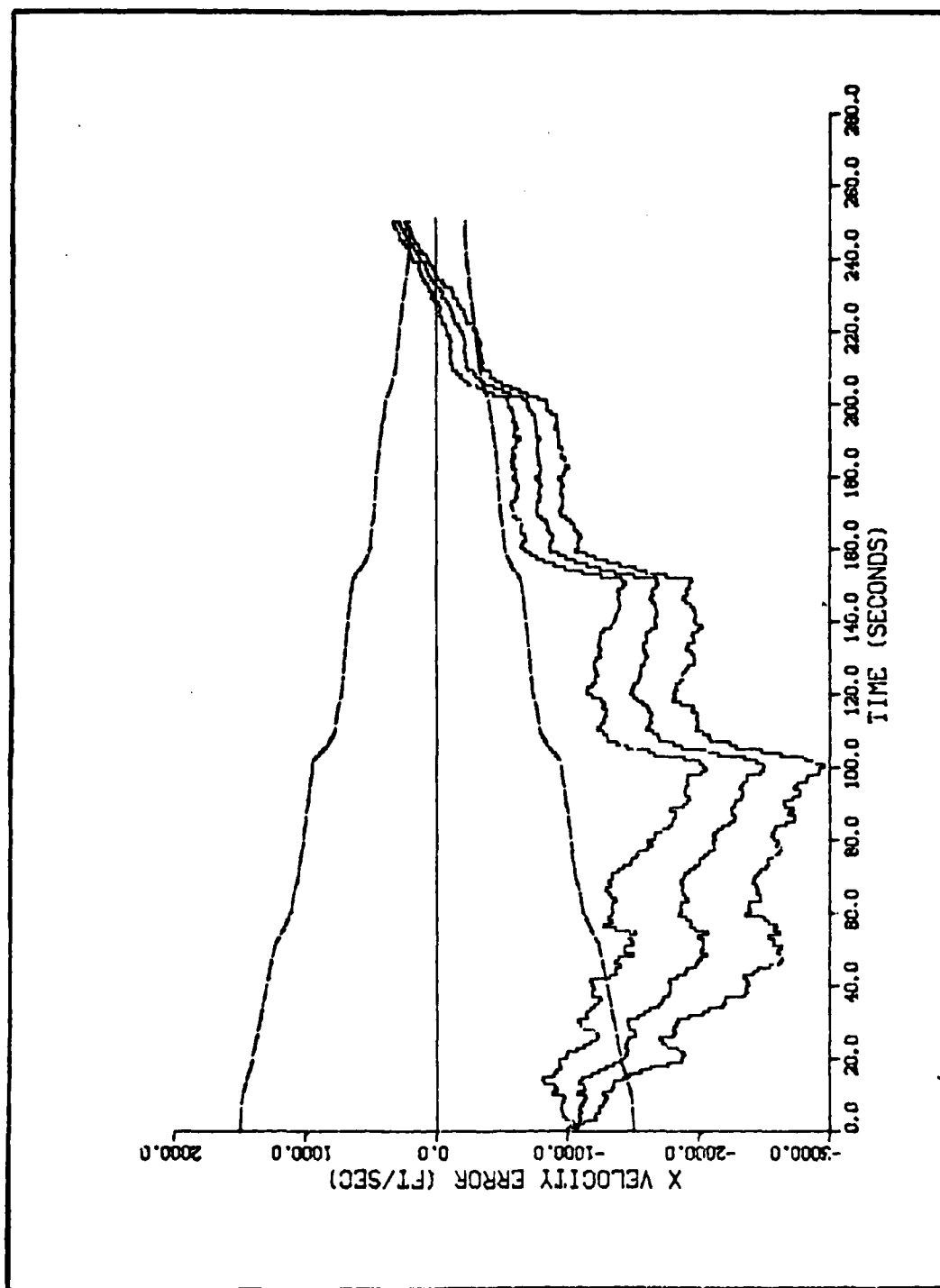


Figure O-31. Algorithm 5, Velocity Error in the X Direction Versus Time,
Target Profile 7, $\sigma_{\text{ANGLE}} = 1 \text{ mrad}$

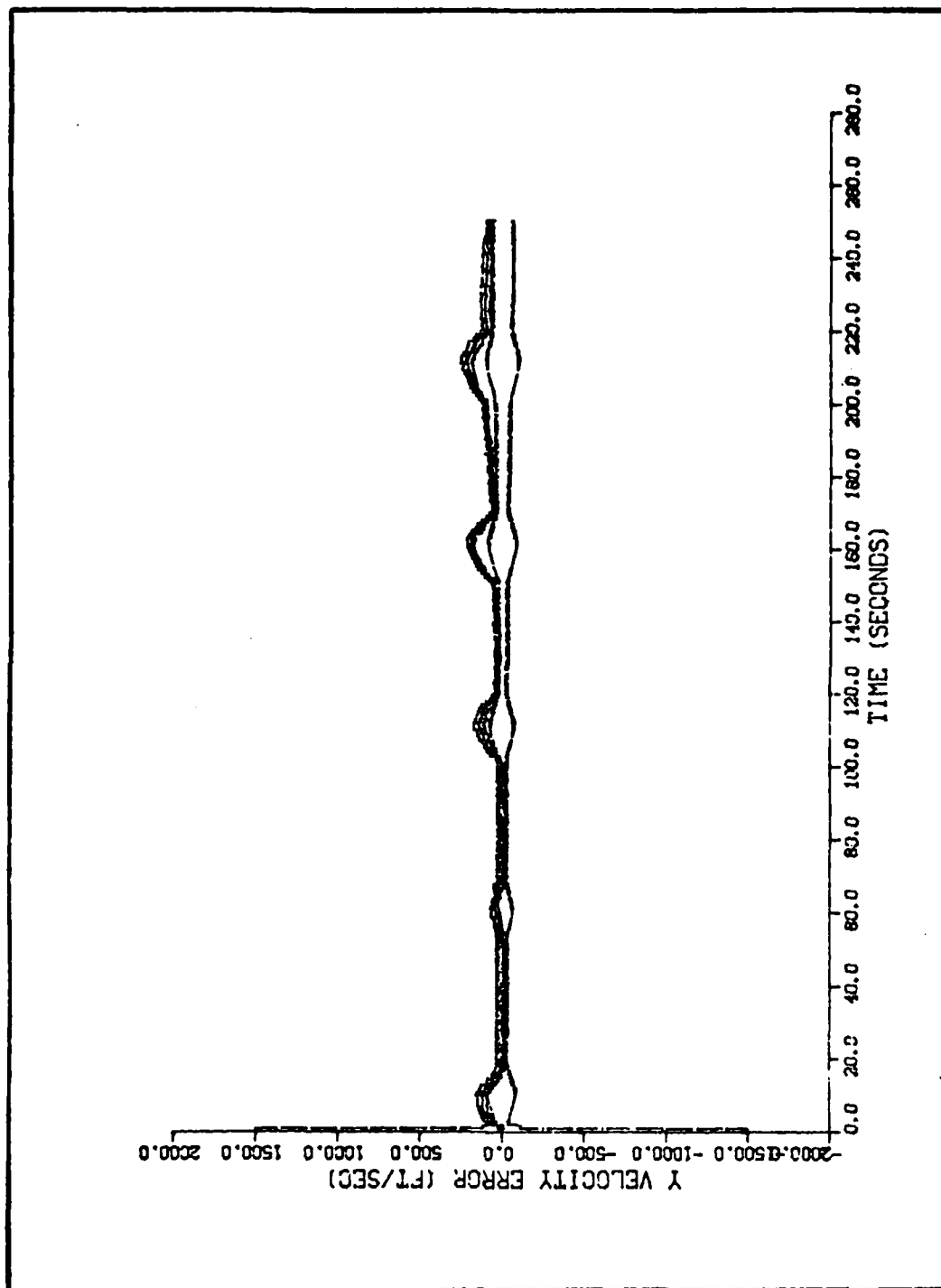


Figure O-32. Algorithm 5, Velocity Error in the Y Direction Versus Time,
Target Profile 7, θ ANGLE = 1 mrad

APPENDIX P
VALUES FOR $\underline{P}(t_0)$

This appendix contains a table which presents all of the values used in the diagonal $\underline{P}(t_0)$ matrix for all the algorithms.

Table P-1 Values for $\underline{P}(t_0)$

	ALGORITHM NUMBER					
	1	2	3	4	5	6
$P_{11}(t_0)$	10^{12}	10^{12}	10^{12}	10^{12}	10^{12}	10^{12}
$P_{22}(t_0)$	2.25×10^6	10^6	10^8	10^6	2.25×10^6	2.25×10^6
$P_{33}(t_0)$	10^3	10^6	10^8	10^{12}	2.25×10^6	2.25×10^6
$P_{44}(t_0)$	4×10^2	2.25×10^6	2.25×10^6	10^6	10^4	10^4
$P_{55}(t_0)$	10^3	2.25×10^6	2.25×10^6	-	-	-
$P_{66}(t_0)$	4×10^2	10^4	2.25×10^6	-	-	-
$P_{77}(t_0)$	10^3	10^3	-	-	-	-
$P_{88}(t_0)$	-	10^3	-	-	-	-
$P_{99}(t_0)$	-	10^3	-	-	-	-

APPENDIX Q

DISCUSSION OF THE BAYESIAN FILTERING APPROACH

The Bayesian filtering approach discussed in Reference 12 was investigated briefly. This approach discretizes the probability density function over a grid in n -space, where the state vector is of dimension n . The propagation cycle propagates each of these state space grid points according to the dynamics equations; the update cycle changes the probability associated with each of the grid points according to a function of the measurement residual (which must be calculated for each grid point). Dynamics noise has the effect of generating new grid points in the area of highest probability. Since there is no error covariance matrix, there is no need to solve the Ricatti equation on line. This method is also supposed to alleviate the divergence problem which, according to Dr. Sorenson, plagues the extended Kalman filter in some applications such as bearings-only ranging.

The biggest problem found with this method was in maintaining the number of grid points at an acceptably low level. In the bearings-only tracking example shown in Reference 12, it is assumed that no dynamics driving noise is being introduced into the system. It is therefore possible to track the target without generating new grid points, thus making the limiting of the number of points relatively simple.

If however, new grid points are generated by dynamics driving noise, several problems arise immediately. The first of these problems is the generation of the new points; that

is, where to place them and how to assign a probability to them. After they are generated, not only do the points with very low probability have to be discarded, but some method must be found for combining the points that are "close together" in n -space. If this were not done, it would be possible for points with high associated probability to break into enough new points with smaller probabilities that all of these new points would be discarded as improbable, even though the probability density of the region in question may not have changed appreciably.

One method examined for accomplishing this was to propagate the grid points using no dynamics noise and reinitialize occasionally about either the "center of mass" of the old system of grid points or the point with the highest associated probability. This involves picking a single point and establishing a new grid in n -space with the chosen point as the center. The question now becomes how to reassign the probabilities so that the probability density information contained in the old grid is not completely disregarded. It was at this point that efforts in the Bayesian method were abandoned; however, this approach is worth pursuing if time is available.

VITA

Donald R. Stiffler was born [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]. He graduated from [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]. Upon graduating, he entered the Air Force as a cadet at the United States Air Force Academy, from which he earned a Bachelor of Science in Electrical Engineering degree in 1974. He next entered undergraduate pilot training at Moody Air Force Base, Georgia, and was awarded his pilot wings in 1975. He was a instructor pilot in T-38 aircraft at Columbus Air Force Base, Mississippi, and an aircraft commander in F-15 aircraft at Holloman Air Force Base, New Mexico, prior to entering the Air Force Institute of Technology in 1981.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/82D-64	2. GOVT ACCESSION NO. AD-A125405	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ANALYSIS OF SIX ALGORITHMS FOR BEARINGS ONLY RANGING IN AN AIR-TO-AIR ENVIRONMENT		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald R. Stiffler Captain		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Superiority Group (AFWAL/AART-1) Air Force Wright Aeronautical Laboratory Wright-Patterson AFB, Ohio 45433		12. REPORT DATE December 1982
		13. NUMBER OF PAGES 272
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES <div style="text-align: center;"> <p>Approved for public release: IAW AFR 190-17.</p> <p><i>Lynn E. Wolaver</i></p> <p>LYNN E. WOLAVER</p> <p>Dean for Research and Professional Development</p> <p>Air Force Institute of Technology (AIC)</p> <p>Wright-Patterson AFB OH 45433</p> </div> <div style="text-align: right;"> <p>4 JAN 1982</p> </div>		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Passive Ranging Bearings-Only Ranging		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Six algorithms for finding the range to an airborne target using noise-corrupted bearing and bearing rate measurements from an airborne tracker were developed and evaluated. The algorithms all employ extended Kalman filters, one using a spherical coordinate system representation, and the other five using Cartesian system representations. The target models include independent, Gauss-Markov accelerations (two algorithms), constant turn rate motion in a single plane (one algorithm), and		

basically straight and level flight (three algorithms).

The algorithms were evaluated in a Monte Carlo analysis. The truth model for this analysis was based upon two trajectories, one for the target and one for the attacker, which were deterministically set before the Monte Carlo runs. The output of the simulation was composed of plots of the error process, including the mean of the errors, the mean of the errors plus and minus one standard deviation, and the filter-generated estimate of the standard deviation of the errors.

It was found that, due to an observability problem, only the algorithms which modelled target motion as straight and level flight worked satisfactorily. These algorithms worked best in scenarios in which the actual target motion was nearly directly toward or directly away from the attacking aircraft.